



GE VERNOVA

DIGITAL

PROFICY Workflow

User Guide

Proprietary Notice

The information contained in this publication is believed to be accurate and reliable. However, General Electric Company assumes no responsibilities for any errors, omissions or inaccuracies. Information contained in the publication is subject to change without notice.

No part of this publication may be reproduced in any form, or stored in a database or retrieval system, or transmitted or distributed in any form by any means, electronic, mechanical photocopying, recording or otherwise, without the prior written permission of General Electric Company. Information contained herein is subject to change without notice.

© 2023, General Electric Company. All rights reserved.

Trademark Notices

GE, the GE Monogram, and Predix are either registered trademarks or trademarks of General Electric Company.

Microsoft® is a registered trademark of Microsoft Corporation, in the United States and/or other countries.

All other trademarks are the property of their respective owners.

We want to hear from you. If you have any comments, questions, or suggestions about our documentation, send them to the following email address:
doc@ge.com

Contents

- Chapter 1. Workflow IPI..... 13**
 - About the IPI..... 13
 - Release Notes..... 13
 - Upgrade Workflow..... 20
 - Overview.....20
 - Upgrading Workflow.....23
 - Software Requirements.....25
 - Application Server Software Requirements..... 25
 - Client-only Software Requirements..... 30
 - Reporting Database Software Requirements..... 31
 - Web Task List Software Requirements.....31
 - Workflow Task Client Software Requirements..... 32
 - Server Clustering and Failover Software Requirements..... 32
 - Extension (Workflow and User) Servers Software Requirements..... 33
 - Hardware Requirements.....33
 - Hardware Recommendations.....33
 - Hardware Requirements for an Application Server.....36
 - Hardware Requirements for Single Servers.....37
 - Hardware Requirements for Extension Servers.....37
 - Hardware Requirements for User Servers..... 38
 - Hardware Requirements for Clients..... 38
 - Hardware Requirements for a Reporting Database..... 38
 - Server Clustering and Failover Hardware Requirements..... 39
 - Compatibility..... 39
 - Compatibility and Upgrading with GE Products..... 39
 - Compatibility Issues with Workflow..... 47
 - Changes to Functionality and Supporting Technology 48

Chapter 2. Get Started.....	50
Get Started Guide.....	50
Installation Requirements.....	50
Pre-Installation Consideration.....	54
Installation Procedures.....	79
Server Clustering Implementation.....	103
Deploy the Web Task List.....	116
Log On Overview.....	126
Post-Installation Configuration.....	133
Upgrade Workflow.....	161
Install a Reporting Database.....	164
Terminal Server.....	172
ActiveX Task Controls.....	182
Workflow Task Client.....	185
Installation Security.....	210
Chapter 3. Tutorials.....	215
Tutorials.....	215
Configure access to specific equipment instance details.....	215
Developing a web form for a workflow.....	217
Creating unique displays, solutions, models.....	222
Create a work segment response for a work request using a workflow.....	226
Write work data for a work segment response using a workflow.....	227
Configure forms to view during run time after they have been completed or acquired.....	229
Configure a database statement for a SQL service provider.....	231
Configure multiple servers on a single machine.....	233
Chapter 4. Orientation.....	236
Workflow Overview.....	236
Orientation.....	236
Multiple Sessions of Workflow.....	236

User Interface.....	236
User Interface.....	236
Resources.....	244
Resources.....	244
Search for Resources.....	245
Searching for Resources.....	245
Expression Editor.....	251
Expression Editor.....	251
Chapter 5. Configure Your Environment.....	292
User Password Configuration.....	292
Workflow User Password Configuration.....	292
Computer Management.....	294
Computer Management.....	294
User Interface Controls in the Task List.....	311
User Interface Controls in the Task List.....	311
Custom Solutions and Displays.....	316
Custom User Solutions and Displays.....	316
SSL Encryption for SQL Server.....	321
SQL Server SSL Encryption.....	321
Certificate Handling.....	323
OPC UA Certificate Handling.....	323
Generate Certificates for Workflow 2.6.....	328
Modify Server Certificate Folder Location.....	328
Resolve Port conflicts between prior Proficy Workflow installations and Web HMI.....	329
Chapter 6. Security.....	331
Security Overview.....	331
Workflow Security.....	331
Understanding Security.....	331
Users and Groups.....	338

Workflow Users and Groups.....	338
Domain-Based Security.....	357
Windows Domain-Based Security.....	357
Chapter 7. Resource Information and Configuration.....	407
Resource Information and Configuration.....	407
Resource Information and Configuration.....	407
Material Model.....	408
Material Model.....	408
Material Inventory.....	417
Material Inventory.....	417
Equipment Model.....	421
Equipment Model.....	421
Production Model.....	454
Production Model.....	454
Add properties or parameters to a resource.....	455
Modify the name and description of a property or parameter.....	455
Modify the values for a property or parameter.....	456
Delete properties or parameters from a resource.....	456
Work Data Classes.....	457
Work Process Segments.....	461
Work Definition Segments.....	477
Work Definitions.....	486
Work Requests.....	498
Work Requests View.....	513
Events Model.....	517
Events Model.....	517
Event View.....	555
Event View.....	555
Actuals.....	557

Actuals.....	557
Property Groups and Aspects.....	559
Property Groups.....	559
Aspects.....	559
Chapter 8. Data Administration.....	560
Data Administration Introduction.....	560
Email Server Configuration.....	560
Proficy System.....	561
Electronic Signatures and Audit Trails.....	562
Electronic Signatures and Audit Trails.....	562
Purpose of Electronic Signatures.....	562
Data Sources.....	582
Data Sources.....	582
Archives.....	625
Archives	625
Custom Property Types.....	629
Custom Property Type Editor	629
Transferring Workflow.....	631
Transferring Workflow Configurations.....	631
Transferring Different Types of Configurations Together.....	634
Transferring Configurations by Resource Type.....	652
Equipment Provisioning.....	658
Reporting.....	662
Reporting.....	662
Chapter 9. Forms and User Displays.....	699
Forms Introduction.....	699
Forms and User Displays Authoring Guide.....	699
Designer Panels.....	710
Designer Panels.....	710

Designer: Project Panel.....	712
Designer: Document Tree.....	716
Designer: Control Toolbox.....	725
Designer: Resources.....	1098
Designer: Properties Panel.....	1099
Navigate URI.....	1099
Designer: Parameters Panel.....	1103
Designer: Data Items Panel.....	1108
Designer: Form Events Panel.....	1112
Designer: Server Methods.....	1119
Designer: Client Methods.....	1126
Server and Client Methods Events.....	1147
Designer: Test Panel.....	1152
Designer: Default Properties.....	1153
Designer: Preview Panel.....	1155
Display and Group Duplication.....	1156
Display and Group Duplication.....	1156
Form Templates.....	1160
Form Templates.....	1160
Basic Forms and Displays.....	1177
Basic Forms and Displays.....	1177
Web-Based Forms and Displays.....	1184
Web-based Forms and Displays.....	1184
Existing Assembly Options.....	1221
Existing Assembly Options.....	1221
Form Development in Visual Studio.....	1224
Form Development in Visual Studio.....	1224
Form Development in Silverlight.....	1241
Forms Development in Silverlight.....	1241

Chapter 10. Building and Configuring Workflows.....	1284
Workflow Design Basics.....	1284
Workflow Design Basics.....	1284
Workflow Foundations.....	1298
Workflow Foundations.....	1298
Task Management for Administrators.....	1307
Task Management for Administrators.....	1307
Task Configuration.....	1313
Task Configuration.....	1313
Task Step Configuration.....	1370
Task Step (Subprocess) Configuration.....	1370
Global Subprocesses.....	1411
Global Subprocesses.....	1411
Make Global/Local.....	1415
Basic Activities.....	1416
Activities Used to Create Workflows.....	1416
Third-party Activities.....	1422
Predefined User Activities.....	1426
Data Access Activities.....	1433
Workflow Control Activities.....	1451
Message Queue Activities.....	1470
Flow Control Activities.....	1478
Event-based Activities.....	1488
Keyword Activities.....	1494
Operator Activities.....	1496
eSOP Activities.....	1526
Schedules.....	1530
Schedules.....	1530
Faults.....	1555

Faults.....	1555
Test Workflows.....	1564
Testing Workflows.....	1564
Debug Workflows.....	1566
Debugging.....	1566
Track Data.....	1573
Tracking Data in Workflows.....	1573
Web Services Service Provider.....	1576
Web Services Service Provider (WSSP).....	1576
Legacy Workflow Web Service.....	1585
Legacy Workflow Web Service.....	1585
Categories.....	1592
Categories.....	1592
Completion Codes.....	1595
Completion Codes.....	1595
Actuals.....	1599
Actuals.....	1599
REST APIs.....	1608
REST APIs.....	1608
Chapter 11. eSOP.....	1625
eSOP Orientation.....	1625
eSOP Orientation.....	1625
Client Layout Configuration for eSOP Users.....	1631
Client Layout Configuration for eSOP Users.....	1631
Design eSOP Workflows Overview.....	1647
Design eSOP Workflows Overview.....	1647
Task Construction for Procedures.....	1664
Task Construction for Procedures.....	1664
Task Maintenance.....	1676

Task Maintenance.....	1676
Chapter 12. ActiveX Task Controls.....	1689
Home.....	1689
ActiveX Task Controls.....	1689
Task Controls in GE HMIs.....	1689
Install the ActiveX Task List.....	1689
Modify the ActiveX Task List installation configurations.....	1690
Task List Configuration.....	1691
Task List in HMIs.....	1694
Task Indicator in HMIs.....	1697
Chapter 13. Service Interfaces.....	1701
Service Interfaces.....	1701
HTTPS Web Services.....	1701
User Permissions Between a Workflow and Service Provider.....	1703
Workflow Methods.....	1704
Workflow Methods.....	1704
IArchiveManagement.....	1705
IAuditTrail.....	1707
IDatabaseService.....	1708
IDirectorySearch.....	1712
IEmailService.....	1723
IEquipment.....	1730
IEquipmentConfig.....	1764
IEquipmentHistorian.....	1764
IEvent.....	1772
IExtensionData.....	1773
IITService.....	1774
ILicense.....	1776
IMaterial.....	1777

IopcClient.....	1825
IopcUa.....	1829
IPersonnel.....	1831
IProductionModel.....	1850
IProductionReport.....	1877
IProductionRuntime	1906
IProficyHistorian.....	1982
IPropertyGroup.....	1991
IResourceLinking.....	1995
iFix Service Provider.....	2006
iFIX Service Provider.....	2006
Chapter 14. Operator Task List in the Workflow Client.....	2007
Task List Overview.....	2007
Using the Task List.....	2007
General Operating Environment.....	2007
General Operating Environment.....	2007
General Task Information.....	2023
General Task Information.....	2023
Equipment Context.....	2025
Equipment ContextShow Tasks for Equipment.....	2025
Forms View.....	2027
Forms View.....	2027
Documents.....	2031
Documents.....	2031
Operator Work Instructions.....	2034
Operator Work Instructions.....	2034
Advanced Operator Tasks.....	2036
Advanced Task Actions for Operators.....	2036
Task List Filter Actions.....	2049

Task List Filtering.....	2049
Start Task.....	2065
Start Task.....	2065
Completed Task Information.....	2068
Completed Task Information.....	2068
Electronic Signatures in the Task List.....	2069
Electronic Signatures in the Task List.....	2069
Chapter 15. Best Practices.....	2078
Installation.....	2078
Installation.....	2078
Workflow Performance.....	2078
Workflow Performance.....	2078
Persistence.....	2078
Memory.....	2080
Tracking Configuration.....	2080
Subprocesses.....	2082
Start Workflow and Start Schedule Activities.....	2083
While and For Each Activities.....	2083
Write Activity.....	2085
If/Else Activity.....	2086
Form Events.....	2088
Forms.....	2088
Forms.....	2088
Customize Workflow.....	2089
Workflow Customization.....	2089
Index.....	

Chapter 1. Workflow IPI

About the IPI

The IPI (Important Product Information) provides a quick reference to what's new in the latest version of Workflow.

The following information is available for Workflow 2.6

Category	Description
What's New or Changed in This Release	This section outlines the new and updated product features that are available for this release. Links are available to relevant documentation.
Recommendations	Recommendations provide procedures that may be useful in specific circumstances. Links are available to relevant documentation.
Limitations	Limitations provide notifications about possible application and system constraints or restrictions in this release.
Resolved Issues	Resolved issues are those defects that have been fixed for the release.
Known Issues	Known issues are those issues that are documented in the defect tracking system that affect this release, and may be fixed in a future release.

Release Notes

Provides a list of new and updated features for this release, as well as any resolved issues, limitations, and known issues.

Beginning with Workflow 2.6, information is available on a single page with links to relevant documentation. Use the following links to navigate to each section on this page.

- [What's New or Changed in This Release \(on page 14\)](#)
- [Recommendations \(on page 14\)](#)
- [Limitations \(on page 15\)](#)
- [Resolved Issues \(on page 15\)](#)
- [Known Issues \(on page 19\)](#)

What's New or Changed in This Release

Workflow 2.6 SP2

This section outlines the new and updated product features for the 2.6 SP2 release.

- SQL Server 2022 is supported for Workflow. See:
 - [SQL Server Options \(on page 26\)](#)

Workflow 2.6 SP1

This section outlines the new and updated product features for the 2.6 SP1 release.

- Install Workflow Task Client from the `InstallFrontEnd.exe` file. The installer adds the task client plugin to Operations Hub. See:
 - Install the Task Client in the *Getting Started* Guide.
- Workflow is updated to the latest released version of the Common Licensing software, v18.9.
- SQL Server 2019 is supported for Workflow. See:
 - [SQL Server Options \(on page 26\)](#)
- Workflow is updated with the latest End User License Agreement (EULA) and copyright stamp, where applicable.
- Workflow is migrated to the Visual C++ 2015-2019 Redistributable package for both 32-bit and 64-bit operating systems.
- Workflow is integrated with the Web Help system to provide help documentation, which is accessed using any regular browser.

Recommendations

Recommendations provide procedures that may be useful in specific circumstances.

- When installed on the same machine, GE Web HMI and Workflow both require port 8444 for installations prior to Workflow 2.6. The Workflow 2.6 default port is now 8447, however during an upgrade from versions earlier than 2.6, the port number is NOT changed, and remains 8444. For information about resolution of port conflicts, see [Resolve Port conflicts between prior Proficiency Workflow installations and Web HMI \(on page 329\)](#).
- Upgraded Workflow 2.6 installations from earlier versions continue to use the existing configured certificates, regardless of the operating system. If you are upgrading or installing Workflow on an older operating system and you want certificates with SAN properties, you can use the Workflow ConfigureCertificates tool to generate new certificates. See [Install Certificates on older Operating Systems \(on page 62\)](#) to generate certificates with the SAN property.

- You can now configure Workflow to connect to a remote SQL Server instance using Windows Authentication. Prior to Workflow 2.6, it was mandatory to configure the Workflow SQL Server instance to use Mixed Mode Authentication. This requirement has been removed from Workflow 2.6 and it is now possible for Workflow to connect to a SQL Server instance configured for Windows Authentication only. For security reasons, Windows Authentication is recommended as it eliminates the need for Workflow to save the database user name and password to disk. See [Configure Windows Authentication for Workflow \(on page 56\)](#).

Limitations

Limitations provide notifications about possible application and system constraints or restrictions in this release.

- Beginning with the Workflow 2.6 release, Vision is not shipped as a part of Workflow. The Vision help topics have not been updated, and only apply to prior Workflow versions. Note that Vision 6.3 can be installed separately and will function with Workflow 2.6.

Resolved Issues

Resolved issues are those defects that have been fixed for the current release.



Note:

These fixes are ported to 2.6 SP2 from 2.6 SP1, 2.5 SP4 SIM 5, SIM 6, and SIM 7 releases.

Workflow 2.6 SP2 Fixes:

Tracking ID	Description
DE161690	Installation of Microsoft Visual C++ 2015-2019 Redistributable Package (x86) failed when installing Workflow Task Client 2.6 SP1.
01085838	Proficy Workflow 2.6 SP2 could not connect to the Historian server.
01026968	When loading Workflow Task Client in Operations Hub 2022, a white rectangular border appeared below the Task Client page.
01109095	Equipment Model did not filter in Operations Hub task list.

Workflow 2.6 SP1 Fixes:

Tracking ID	Description
US421239	The method <code>DeleteMaterialLotOrSublotWithActuals</code> introduced in 2.5 SP4 SIM4 has been removed from the IMaterial interface. This method did not perform consistently in all cases.
DE136625	Report synchronization failed due to a duplicate key issue in the <code>Dim_Workflow-Schedule</code> and <code>Dim_SegReqEquipmentSpecification</code> packages. To resolve the issue, data staging queries for the two reporting packages have been modified to exclude any duplicate records.
DE136672	There were no alerts or reminders for security certificates that are expiring within a month. This issue has been resolved. Workflow certificates are checked daily for their expiration period. If any certificate is due for expiry within a month, a warning message is displayed to the logged in user.
DE136677	Errors occurred when saving class changes for model objects due to performance issues. This issue has been resolved.
DE136635	<p>Resolved performance issues related to export and import of materials in a model, and also browsing these materials on the Client:</p> <ul style="list-style-type: none"> • For Export: Fixed the redundant data creation, and optimized the export logic. • For Import: Improving the export performance resulted in less number of excel files for import. Import logic is also optimized while saving the GUIDs in excel.
DE136626	Workflow instances stopped working due to <code>System.Data.SqlClient.SqlException</code> and had to be terminated. Handled the specific exception by retrying workflow instance completion three more times. This allowed for an extra time frame of total 180 (60*3=180) seconds to resume the network.
DE136627	After restarting the Proficiency Server, Workflow calendar based time events do not occur if the event's first occurrence date is in the past. This issue is resolved by setting the <code>Repeat event every ___ day</code> field value appropriately. It is set to 1 if the recurrence is needed daily.
DE136631	Export/Import of user solutions failed without throwing any error messages. To fix this issue, a validation logic has been added to check all the dependent displays of the user solution that are added in the content.

Tracking ID	Description
DE136628	The communication object could not be used for communication. To fix the issue, the <code>MaxMessageSize</code> attribute needs to be configured in <code>ProficiencyClient.exe.config</code> and <code>ProficiencyServer.exe.config</code> by increasing the value from 10485760 to 15728640. The <code>MaxItemsinObjectGraph</code> quota is increased from 300000 to 1500000.
DE136630	Synchronizing tags from Historian to Workflow failed due to conflicts in SQL database collation. This issue has been resolved.

Workflow 2.6 SIM 1 Fixes:

Tracking ID	Description
US274688	The Full Name field can be entered for a Workflow user to display assigned task user's full name in the Web HMI Task List. This field is used when displaying assignment and performer information in the Web HMI Task List.
US287276	Enabled to import user's full name when synchronizing users from the Active Directory. A user's full name is retrieved from the domain controller during the authentication process, and is used to populate the Full Name field on the created user account.
DE9510	Previously, you were not able to duplicate the jumps defined in the eSOP task steps from the Task Editor screen. This issue has been resolved.
DE15101	Previously, on duplicating an eSOP task, the email configuration was not duplicated correctly. To fix this, all email configuration is now preserved when duplicating an eSOP task.
DE47848, DE82901	When importing model objects with aspects applied (such as when Plant Applications is used in conjunction with Workflow), it lead to import failure. The import process is fixed now to handle null values in the import file.
DE49185	Equipment provisioning script created records with invalid parents. To resolve the issue, equipment creation commands are now validated for proper parent-child relationships before creating the equipment.
DE50264	Equipment provisioning script "create property" command (6) created property with NULL data type. To resolve this issue, additional checks have been added to ensure that a valid property data type has been specified.

Tracking ID	Description
DE52998	Communication exception occurred due to a faulty channel state. As a fix, the communication channels are reclaimed if they are detected to be expired, or in a non-opened state.
DE64489	The workflow instances screen displayed an error whenever a workflow instance in the 'Stopped' state was selected. This issue introduced in 2.5 SP3 has been resolved.
DE68886	<p>Projects and Import/Export functions were allowing to import duplicate process segment specifications. To resolve this issue:</p> <ul style="list-style-type: none"> • Database schema has been modified to prevent creation of process segment specifications with a duplicate S95Id. • Projects and Import/Export functions have been modified to fail and display a meaningful error message when an attempt is made to import a process segment specification with a duplicate S95Id.
DE79117	Previously, an exception was thrown when editing a Configure Event Activity. This issue has been resolved.
DE79380	Previously, keywords were not replaced in task level work instructions on the desktop client. This issue has been resolved. Keywords are now replaced as expected in the initial load of work instructions.
DE79400	The Edit Database connection screen was incorrectly limiting port number range, and created incorrect connection string when the port is specified. This issue has been resolved. The valid range for the SQL Server port number is now 1 to 65535. The data source in the connection string excludes the instance name if a port number is also specified.
DE79401	Previously, you were not able to save the database connection with Excel ODBC data provider DSN. This error happened due to a missing database (worksheet) property on the DSN. The error message has been updated.
DE82900	Exporting an equipment failed for the equipment that contained the property with ' (single quote) in its name. The S95 editor also failed to show any properties for the equipment instance when such a property was created. This issue has been resolved.

Tracking ID	Description
DE82902	Assigning tasks dynamically made the tasks visible to incorrect users. This issue has been resolved. Assignment context is now correctly determined when the assigned resource is not set until runtime.
DE89213	Previously, indexes duplicating the primary key existed on many SOADB tables. The duplicate indexes are no longer created, and existing duplicate indexes will be removed during an upgrade from earlier versions.
DE83062	Previously, importing personnel class failed if the target system was missing aspects from the source system. Personnel objects can now be imported successfully when the target system does not have all personnel model aspects present on the source system.
DE83916	Configure Server was not updating the port numbers in the <code>UserStartup.dita</code> file. This issue has been resolved. If the server HTTP or TCP ports are modified using the Configure Server tool, the local client configuration is also updated to connect on the new ports.
DE91292	You can now delete a lot or subplot and all associated material actuals from the production responses. A new method has been added to the IMaterial interface: <code>DeleteMaterialLotOrSublotWithActuals</code> . Use this method with caution!
DE91394	Instance class cannot be found during process to create or update material definition properties. This issue has been resolved.
DE95051	The default height of the navigator did not facilitate the visibility of all the models. The navigator has been fixed with more vertical space to display all the default models without the need to scroll through them.
DE95754	The Generate Web HMI Model process failed with the following error: "Unable to cast object of type 'System.DBNull' to type 'System.String' on Equipment instances with null descriptions." This issue has been resolved to correctly process the equipment instances with null description fields.

Known Issues

Known issues are those issues that are documented in the defect tracking system that affect this release, and may be fixed in a future release.

Tracking ID	Description
DE8399	<p>Browsing OPC Items from RSLinx Classic OPC Server - When browsing OPC items from the server, branches show items from previously expanded branches.</p> <p>Workaround: Reset the browser by clicking the Browse Server button again, then browse the desired branch.</p>
DE56348	<p>After an OPC connection to a Cimplicity server is added in Workflow, new projects added to the Cimplicity OPC server are not visible from the OPC Item browser in Workflow.</p> <p>Workarounds:</p> <ul style="list-style-type: none"> • Change the ProgID of the OPC connection to a different value and Save. Then change the ProgID back to the original value and Save. • Delete the OPC server connection from Workflow and re-create it. • Restart Proficy Server. <p>Performing any of the above actions will cause the correct list of items to be returned from the Browse action.</p>
DE76343	<p>After installation on Windows 10 or Window Server 2016 "General Electric" Start Menu has no content.</p> <p>This issue is caused by Microsoft's Windows operating system.</p> <p>Workaround: Log out of your current Windows session and log back in.</p>
DE136938	<p>Guest user privileges updated in Workflow are not getting updated to Operations Hub task client. As a result, when guest users with no privileges are upgraded to possess administrative rights in Workflow, they still are not able to access their scheduled tasks in the task client widget in Operations Hub.</p> <p>Workaround: Restart Proficy Server.</p>

Upgrade Workflow

Overview

**Important:**

When you upgrade the application server, you must also upgrade all of the remote clients that connect to that server to use the same version of Workflow. You can upgrade Workflow without having to uninstall the program.

Upgrade Paths

The following upgrade paths are supported:

- From Workflow 2.5 with latest SIM to Workflow 2.6
- From Workflow 2.5 SP1, SP2, SP3, SP4 with latest SIM to Workflow 2.6
- From Workflow 2.5 Reporting SP1, SP2, SP3, SP4 with latest SIM to Workflow 2.6
- From Workflow 2.2 SP1, SP2 to Workflow 2.6
- From Workflow 2.2 SP1, SP2 Reporting to Workflow 2.6 Reporting
- From Workflow 2.5 to Workflow 2.5 SP1

**Note:**

All versions of Workflow prior to version 2.5 SP1 must first be upgraded to version 2.5 before you can upgrade to Workflow 2.5 SP1.

- From Workflow 2.1 to Workflow 2.5
- From Workflow 2.2 service pack 1 (SP1) to Workflow 2.5
- From Vision 6.2 to Vision 6.3

**Note:**

All versions of Vision prior to version 6.3 must be uninstalled before being upgraded to Workflow 2.5 SP1.

- From Workflow 2.1 to Workflow 2.2 service pack 1 (SP1)
- From Workflow 2.0 to Workflow 2.2 service pack 1 (SP1)
- From Workflow 1.5 service pack 4 (SP4) to Workflow 2.2 SP1

**Note:**

You must first upgrade Workflow 1.5 SP4 to Workflow 2.0.

If you are upgrading from an earlier version of Workflow, contact Technical Support.

Legacy Certificates

Starting with Workflow 2.5, legacy certificates are no longer supported.

Upgrading Extension Servers

If you are installing or upgrading an extension server (Workflow or User), you must have already installed or upgraded an application server of the same version to host the Core services on a separate machine.

Upgrading Microsoft® .NET Framework 4.5 (Full Framework)



Note:

Custom display or form assemblies that target earlier versions of .NET will continue to function as before. However, saving changes made to such custom assemblies requires that they be upgraded, which is accomplished in different ways, based on the version of .NET:

- For an assembly that pre-dates .NET 4.0, you must upgrade the assembly when you open it. Failure to do so results in an error and the inability to save the edited assembly.
- For an assembly that targets .NET 4.0, the assembly is upgraded automatically when opened, and changes to such assemblies can be saved as before.

Service providers that target versions of the .NET framework before version 4.5 must be recompiled targeting .NET 4.5.

Upgrading Web-based Forms for the Task List

If you have web-based forms as part of an existing workflow that you have created prior to the release of Workflow 2.2 SP1, you must bind them again inside the **Form activity** in order for them to function in the Task List.



Note:

This upgrading issue affects existing web-based forms only, and not HTML5 web forms that are created in Workflow 2.2 SP1.

Upgrading a Clustered Environment

After upgrading Workflow on the primary server machine, you must reconfigure the failover server in the cluster, as well as reconfigure that machine as the primary server in the cluster.

SQL Server Database Replication

When upgrading the application, if you are using a replicated database and have replication turned on, you must turn it off, upgrade, and then turn it on again. You must also upgrade the replicated database. For more information, see **Upgrade a Replicated Database**.

Password Security

During an upgrade installation, the account lockout capability is automatically enabled; all other password security features are disabled, by default. To enable any of the other password security features, you must use the **Configure Security** tool after the upgrade installation has successfully completed.

Windows Users

Workflow 2.0 supports Windows domain names with personnel names and login names. If you choose to use the new functionality included with Workflow 2.0 (that is, active directory universal or global groups mapped to Workflow groups), then when your Windows users log in, their personnel name and login name will both be updated to include the domain name, and the personnel name will change to match the login name. For example, the personnel name *John Smith* and login name *johnsmith* will both change to *<domain name>\johnsmith*.



Warning:

If you were using Windows user accounts in a prior version of Workflow, any workflows that reference individual users will no longer work! These personnel names will NOT be updated when the Windows users log in. You must manually change these names to include the domain name with a backslash character between the domain name and personnel name. However, if you reference personnel classes in your workflows, no change is required; the workflows will work as they did in the previous version of Workflow.

Upgrading Workflow

You can upgrade your Workflow software without uninstalling the previous version.

About this task



Note:

- If you are upgrading your database to SQL Server 2019, you must first upgrade Workflow to v2.6 or greater, and then upgrade SQL Server.
- Administrative user names and passwords used during your initial installation are not retained during the upgrade process. You can set these credentials again during this process.
- Starting with Workflow 2.5, legacy certificates are no longer supported.

Procedure

1. Run the installation software. If the splash screen does not appear, run `InstallFrontEnd.exe` on the root directory of the installation folder.
2. Review the license agreement, and then click **I Agree**.
The **Security** page appears.
3. Specify the administrator's authentication credentials.
 - a. In the **Name** field, accept the default name or enter a new name.
 - b. In the **Password** field, enter a password.



Note:

- If password complexity is enabled and configured, the Administrator password must follow the same rules as defined for user passwords.
- If you are using the **Workflow Task List in Web HMI**, select the **Web HMI Access** check box, enter the `Web HMI Host name OR IP address`, and then click **Next** to view the **Application Upgrade** page.



Important:

When you use the Configure Security tool to make changes to password security, you must re-enter and confirm the Administrator user's password before you can save your changes.

The **Application Upgrade** page appears.

4. **Optional:** If you do not want to back up your SQL Server database, or if you have performed your own backup, select the Do not back up the database automatically check box.

**Tip:**

At any time during the configuration process, you can click **Back** to return to a previous page to change your settings.

5. Click **Install**.

The **Installation** page appears, displaying the status of each installation step.

6. When the installation is complete, click **Exit**.**What to do next****Note:**

If the installation fails at any point during the Database setup and configuration, you will have to uninstall Workflow via the Windows Control panel, restore your database from backup, resolve the database issue and perform the upgrade again.

If the installation fails at any point after the Database setup, or you want to change a configuration setting after the installation completes, you can open the appropriate standalone component to configure the required information. For more information, see: [Post-installation Configuration \(on page 133\)](#).

Software Requirements

Application Server Software Requirements

This topic describes the software requirements for your installation.

Installation

**Note:**

In addition to the following material, see the **Architecture Considerations** in the web help for information on options related to the operation mode when installing on a 64-bit computer.

Use the following links to view the various options:

- [Operating System Options \(on page 26\)](#)
- [SQL Server Options \(on page 26\)](#)
- [.NET Requirements \(on page 28\)](#)

- [Other Requirements \(on page 29\)](#)
- [Other Supported Software \(on page 29\)](#)

Related information

[Enable Connection between Workflow and SQL Server \(on page 72\)](#)

Operating System Options

Operating system options apply to all servers that utilize Workflow.

The following operating systems are supported:

- Microsoft® Windows® 10 (32- or 64-bit)
- Microsoft® Windows® 8 (64-bit)
- Microsoft® Windows® 7 Professional SP1 (32- or 64-bit)
- Microsoft® Windows® Server 2022
- Microsoft® Windows® Server 2019 (32- or 64-bit)
- Microsoft® Windows® Server 2016 (32- or 64-bit)
- Microsoft® Windows® Server 2012 R2 (64-bit)



Note:

You must have the April 2014 updated rollup for Windows® Server 2012 R2 (2919355) installed.

- Microsoft® Windows® Server 2012 Standard (64-bit)
- Microsoft® Windows® Server 2008 SP2 (32-bit only)
- Microsoft® Windows® Server 2008 R2 SP1 (64-bit)



Note:

Terminal server environments are supported only on Windows Server 2008 R2 (64-bit)

SQL Server Options

SQL Server options apply to all servers that utilize Workflow.

**Note:**

- The Workflow project is installed into this SQL Server database. SQL Server can be installed on a different computer that is also on the domain or workgroup; however, you must ensure that network configurations allow appropriate machine communication.
- For security reasons, we recommend that SQL Server be configured with Windows authentication, but Mixed Mode authentication is also supported.

The following SQL Server systems are supported:

- Microsoft® SQL Server 2022

**Note:**

If you are upgrading your database to SQL Server 2022, you must first upgrade Workflow to v2.6 SP2 or greater, and then upgrade SQL Server.

- Microsoft® SQL Server 2019

**Note:**

If you are upgrading your database to SQL Server 2019, you must first upgrade Workflow to v2.6 SP1 or greater, and then upgrade SQL Server.

- Microsoft® SQL Server 2017
- Microsoft® SQL Server 2016
- Microsoft® SQL Server 2014 (32- or 64-bit)
- Microsoft® SQL Server 2012 Enterprise (32- or 64-bit)
- Microsoft® SQL Server 2012 Business Intelligence (32- or 64-bit)
- Microsoft® SQL Server 2012 Standard (32- or 64-bit)
- Microsoft® Windows® Server 2012 Essentials

**Note:**

In order for Workflow to connect to any edition of Microsoft® SQL Server 2012 or later, you must configure SQL Server to enable the *sysadmin* role for the applicable login account. For more information, see **Enable Connection between Workflow and SQL Server 2012**.

- Microsoft® SQL Server 2008 R2 with the latest service pack

- Microsoft[®] SQL Server 2008 Standard or Enterprise Edition (32- or 64-bit)
- Microsoft[®] SQL Server 2008 Express



Note:

If you are using SQL Server Express with the Workflow application server, you cannot use the Reporting Database.

The Workflow installation automatically selects the default SQL Server instance. If SQL Server 2012 and SQL Server 2008 are both installed, SQL Server 2008 may be the default instance. In this scenario, if you want to use SQL Server 2012 instead, you must select this instance as the default as part of manually configuring your SQL Server settings.

For more information, see the topic corresponding to the type of installation you will be performing.

.NET Requirements

.NET requirements apply to all servers that utilize Workflow.

The following .NET option is supported:

Microsoft[®] .NET Framework 4.6 (Full Framework) or greater)



Note:

.NET 4.7 (Full Framework) is mandatory if you are using FIPS or TLS 1.2.



Note:

Custom display or from assemblies that target earlier versions of .NET will continue to function as before. However, saving changes made to such custom assemblies requires that they be upgraded, which is accomplished in different ways, based on the version of .NET:

- For an assembly that pre-dates .NET 4.0, you must upgrade the assembly when you open it. Failure to do so results in an error and the inability to save the edited assembly.
- For an assembly that targets .NET 4.0, the assembly is upgraded automatically when opened, and changes to such assemblies can be saved as before.

Service providers that target versions of the .NET framework before version 4.5 must be recompiled targeting .NET 4.5 or greater.

Other Requirements

Other requirements apply to all servers that utilize Workflow.

The following requirements are supported:

- Microsoft[®] Excel 2007 or greater. To use Excel 2007, you must manually install the Primary Interop Assemblies (for Import/Export operations). These assemblies are included in the Excel 2010 installation.
 - <http://msdn.microsoft.com/en-us/library/kh3965hw.aspx>
 - <http://www.microsoft.com/downloads/details.aspx?familyid=59DAEBAA-BED4-4282-A28C-B864D8BFA513&displaylang=en>
 - <http://technet.microsoft.com/en-us/library/cc179091.aspx>
- Microsoft[®] Internet Explorer 7 or greater
- For web-based forms, Microsoft[®] Internet Explorer 10 or greater is required.

**Note:**

Internet Explorer 10 is supported only on the following operating systems:

- Windows[®] 7 SP1
- Windows[®] 8
- Windows[®] Server 2008 R2 with SP1
- Windows[®] Server 2012 R2

- Windows Communication Foundation HTTP Activation must be enabled.
- Microsoft System CLR Types (32-bit only) for Microsoft SQL Server 2016, 2017, and 2019.

Other Supported Software

Other supported software applies to all servers that utilize Workflow.

The following software is supported:

**Note:**

Proficiency Portal and all features associated with this product are no longer supported with Workflow.

- VMWare Server or VMWare ESX Server



Note:

VMWare servers are supported to use any of the supported operating systems.



Important:

Advanced features of ESX Server (that is, VMotion, VMWare VMFS, and Clustering support) are not supported.

- Hyper-V visualization system for x86-64 systems
- Remote Desktop Session Host (Terminal Services)
- Historian 3.5 or greater

Optional Requirements

- Microsoft[®] Silverlight[™] Runtime 5.0 and the Silverlight SDK 5.0



Note:

Silverlight is required to access and develop forms and displays in the Silverlight Designer, and to view Silverlight forms and displays in the Global Viewer.



Important:

Support for Silverlight and Silverlight forms will be discontinued with the next major release of the software.

- Microsoft[®] AD LDS (Active Directory Lightweight Directory Service) or ADAM (Active Directory Application Mode) with SP1 (available from <http://www.microsoft.com/downloads/details.aspx?familyid=9688f8b9-1034-4ef6-a3e5-2a2a57b5c8e4&displaylang=en>)



Note:

For all new deployments, it is recommended you use Simulated Directory instead of AD LDS for increased performance.

Client-only Software Requirements

This topic describes the software requirements for your installation.

A client-only machine contains only the Workflow client application and must connect to another machine that has the application server installed.

**Note:**

If you have a client-only installation, it must be the same version as the server to which it is connected.

The software requirements for a client-only installation are the same as for the application server.

Use the following links to view the various options:

- [Operating System Options \(on page 26\)](#)
- [SQL Server Options \(on page 26\)](#)
- [.NET Requirements \(on page 28\)](#)
- [Other Requirements \(on page 29\)](#)
- [Other Supported Software \(on page 29\)](#)

Reporting Database Software Requirements

This topic describes the software requirements for your installation.

The Reporting database must be installed on the same machine as the SQL Server instance which will hold the database. A remote SQL Server instance is not supported.

Reporting database installation have the same software requirements as the main application server installation.

Use the following links to view the various options:

- [Operating System Options \(on page 26\)](#)
- [SQL Server Options \(on page 26\)](#)
- [.NET Requirements \(on page 28\)](#)
- [Other Requirements \(on page 29\)](#)
- [Other Supported Software \(on page 29\)](#)

Web Task List Software Requirements

This topic describes the software requirements for your installation.

Web Task List installations have the same software requirements as the main application server installation.

Use the following links to view the various options:

- [Operating System Options \(on page 26\)](#)
- [SQL Server Options \(on page 26\)](#)
- [.NET Requirements \(on page 28\)](#)
- [Other Requirements \(on page 29\)](#)
- [Other Supported Software \(on page 29\)](#)

Workflow Task Client Software Requirements

This topic describes the software requirements for your installation.

In addition to the software requirements as the main application server installation, Workflow Task Client installation also requires the following to be installed on the same machine.

- Operations Hub 1.7

Use the following links to view the various options:

- [Operating System Options \(on page 26\)](#)
- [SQL Server Options \(on page 26\)](#)
- [.NET Requirements \(on page 28\)](#)
- [Other Requirements \(on page 29\)](#)
- [Other Supported Software \(on page 29\)](#)

Server Clustering and Failover Software Requirements

This topic describes the software requirements for your installation.

Operating System Options

The following operating systems are supported:

- Microsoft® Windows® Server 2016 (32- or 64-bit)
- Microsoft® Windows® Server 2012 R2 (64-bit)
- Microsoft® Windows® Server 2012 Standard (64-bit)
- Microsoft® Windows® Server 2008 R2 (64-bit)

Other than the supported operating systems, the software requirements for Server clustering and failover installations are the same as for the main application server installation.

Use the following links to view the various options:

- [SQL Server Options \(on page 26\)](#)
- [.NET Requirements \(on page 28\)](#)
- [Other Requirements \(on page 29\)](#)
- [Other Supported Software \(on page 29\)](#)

Related information

[Supported and Validated Cluster Configurations \(on page 103\)](#)

Extension (Workflow and User) Servers Software Requirements

This topic describes the software requirements for your installation.

Workflow and user extension servers have the same software requirements as the main application server installation.

Use the following links to view the various options:

- [Operating System Options \(on page 26\)](#)
- [SQL Server Options \(on page 26\)](#)
- [.NET Requirements \(on page 28\)](#)
- [Other Requirements \(on page 29\)](#)
- [Other Supported Software \(on page 29\)](#)

Related information

[Enable Connection between Workflow and SQL Server \(on page 72\)](#)

Hardware Requirements

Hardware Recommendations

This section describes the recommended hardware specifications which are needed for Workflow to perform well.

The following table provides general guidelines for hardware recommendations based on sizing specifications.

Project Specification	Small System	Medium System*	Large System*
Equipment Model Objects	0 – 500	500 – 2,500	2,500 +

Project Specification	Small System	Medium System*	Large System*
Equipment Model Properties	0 – 5,000	5,000 – 500,000	500,000 +
Material Definitions	0 – 1,000	1,000 – 10,000	10,000 +
Material Lots	0 – 1,000	1,000 – 5,000	5,000 +
Production Model Objects	0 – 500	500 – 1,000	1,000 +
Production Variables	0 – 5,000	5,000 – 10,000	10,000 +
Number of Workflows	0 – 100	100 – 1,000	1,000 +
Running Workflows	0 – 50	50 – 500	500 +
Concurrently Executing Workflows**	0 – 5	5 – 25	25 +
Configured Events	0 – 50	50 – 500	500 +
Events Triggering per Hour	0 – 15	15 – 100	100 +
Work Requests in System	0 – 100	100 – 1,000	1,000 +
Active Work Requests	0 – 25	25 – 500	500 +
Active Clients	1 – 25	25 – 100	100 +
Recommended Server RAM	8 GB	16 GB	32 GB
Recommended Server CPUs	2 CPUs @ 3GHz	4 CPUs @ 3GHz	8 CPUs @ 3GHz

*Many systems do not use all of the capabilities of Workflow, so some projects may match some specifications in the small system column with other specifications in the medium or large system column. Choose the column that more closely matches the targeted system.

**Any running workflow that is waiting for user input in a task form can be considered *paused* and is considered to be a running workflow but not an executing workflow.

Additional Factors for Hardware Recommendations

Special Application Factors

Server recommendations may need to be increased when the application uses intensive processing or significant disk access. Examples of these scenarios include reading and parsing data files from disk, custom logging or writing to output data files, advanced mathematics in code activities, custom service providers, communicating with third party products, and so on.

SQL Server on the Workflow Server

The hardware recommendations above are for servers where the SQL Server is not on the Workflow server system. If you choose to run SQL Server on the Workflow server, then you must add RAM and CPUs to the recommended amounts.

Running on a Virtualized Machine

The hardware recommendations above are for a server that is not running in a virtualized environment. For servers that are running in a virtualized environment such as a VMWare or Hyper-V image, the recommended amounts must be exclusively allocated to the server image and have as much available physical resources on the virtualization server. For example, if a server needs 8GB of RAM, then that much RAM must be allocated to the image, and the virtualization server must have 8GB of physical RAM available for that image. If the image runs out of physical RAM on the virtualization server, performance degrades significantly.

Running Other Applications

All the hardware recommendations are assuming that the machine is dedicated to running the Workflow server. If you are planning to run other applications on the machine, you will need to increase the CPU and RAM requirements appropriately to handle running these applications. This includes other GE Digital applications, such as Historian, CIMPLICITY, or iFIX.

**Note:**

When using Windows® Server 2008 R2 or Windows® Server 2012 as the operating system, keep the following information in mind. Newer Microsoft operating systems have greater hardware requirements than older ones. Windows® Server 2008 R2 and newer use significantly more memory and more CPU. For equivalent performance, add an additional 2 GB of memory and an additional processor.

Additional Recommendations for the SQL Server Machine

For optimum performance, the databases, transaction logs, `tempdb`, non-clustered indexes, and back-ups should be installed on separate hard drives where possible. The following RAID specifications are also recommended:

- SQL database: RAID 0+1, RAID 1 or RAID 5
- SQL transaction logs: RAID 0+1 or RAID 1, and should be a separate physical device
- SQL `tempdb`: RAID 0+1 or RAID 1
- SQL indexes: RAID 0+1 or RAID 1
- Back-ups: RAID 0 (no fault tolerance, but faster writes) or RAID 5

Additionally, the following SQL Server configuration is recommended:

- SQL Server machine: 4 GB RAM (*required minimum*)
- Use Windows Authentication only. For information on how to connect to a remote SQL Server instance using Windows Authentication, please see **Configure Windows Authentication for Workflow** in the Workflow Help documentation.
- Exclude the database files from anti-virus scanning
- Use multiple processors for every installation
- Enable the default backup job or create a maintenance plan
- For `tempdb`, the number of files configured should be based on the number of processors that the server has (for more information, see [Optimizing tempdb Performance](#))

Other Performance Considerations

The guidelines described here are not the only aspects of a system to be considered to ensure optimum performance. Each individual system should be evaluated for other potential sources of performance issues. Examples of other sources of problems may include, but are not limited to, the following:

- Networking throughput between Workflow server and client
- Networking throughput between Workflow server and SQL Server
- Data throughput between the Workflow server and data sources such as OPC servers, Historian, iFIX, and CIMPLICITY.
- The number of client instances active on a single client machine

Hardware Requirements for an Application Server

This topic describes the hardware required for your product installation.

- 4-socket 8 core 32 GB RAM
- Windows Server x64 OS
- 200 GB free hard disk space
- Network interface software for network communications, and TCP/IP network protocol
- SVGA or better color graphics monitor and a 100% IBM-compatible 24-bit graphics card capable of 1024x768 resolution and at least 65535 colors
- XGA or better color graphics monitor and a 100% IBM-compatible 24-bit graphics card capable of 1024x768 resolution and at least 65535 colors
- Two-button mouse with a scroll wheel or compatible pointing device (*such as a touch screen*)

Hardware Requirements for Single Servers

This topic describes the hardware required for your product installation.

Server Deployment

**Note:**

These requirements apply whether you are installing a multiple server configuration (that is a core server and a Workflow server), or a failover cluster.

- 3 GHz Core™ 2 Duo processor with 8 GB RAM (*minimum*)
- 50 GB free hard disk space (*minimum*)
- DVD-ROM drive
- Network interface software for network communications, and TCP/IP network protocol
- One parallel port or free direct-connect USB port
- XGA or better color graphics monitor and a 100% IBM-compatible 24-bit graphics card capable of 1024x768 resolution and at least 65535 colors
- Minimum monitor support
- Two-button mouse with a scroll wheel or compatible pointing device (*such as a touch screen*)

Hardware Requirements for Extension Servers

This topic describes the hardware required for your product installation.

- 3 GHz Core™ 2 Duo processor with 8 GB RAM (*minimum*)
- 50 GB free hard disk space (*minimum*)
- Network interface software for network communications, and TCP/IP network protocol
- One parallel port or free direct-connect USB port

- XGA or better color graphics monitor and a 100% IBM-compatible 24-bit graphics card capable of 1024x768 resolution and at least 65535 colors
- Minimum monitor support
- Two-button mouse with scroll wheel or compatible point device (*such as a touch screen*)

Hardware Requirements for User Servers

This topic describes the hardware required for your product installation.

- 3 GHz Core™ 2 Duo process with 4 to 8 GB RAM (*dependent on how much activity the server is expected to manage*)
- 50 GB free hard disk space (*minimum*)
- DVD-ROM drive
- Network interface software for network communications, and TCP/IP network protocol
- One parallel port or free direct-connect USB port
- XGA or better color graphics monitor and a 100% IBM-compatible 24-bit graphics card capable of 1024x768 resolution and at least 65535 colors
- Minimum monitor support
- Two-button mouse with scroll wheel or compatible pointing device (*such as a touch screen*)

Hardware Requirements for Clients

This topic describes the hardware required for your product installation.

- 3 GHz Core™ 2 Duo processor with 2 GB RAM (*minimum*)
- 20 GB free hard disk space (*minimum*)
- DVD-ROM drive
- Network interface software for network communications, and TCP/IP network protocol
- SVGA or better color graphics monitor and a 100% IBM-compatible 24-bit graphics card capable of 1024x768 resolution and at least 65535 colors
- Two-button mouse with scroll wheel or compatible pointing device (*such as a touch screen*)

Hardware Requirements for a Reporting Database

This topic describes the hardware required for your product installation.

- 3 GHz Core™ 2 Duo Processor with 3 GB RAM (*minimum*)
- 50 GB free hard disk space (*minimum*)
- DVD-ROM drive
- Network interface software for network communications, and TCP/IP network protocol

- One free direct-connect USB port
- XGA or better color graphics monitor and a 100% IBM-compatible 24-bit graphics card capable of 1024x768 resolution and at least 65535 colors
- Two-button mouse with scroll wheel or compatible pointing device (*such as a touch screen*)

Server Clustering and Failover Hardware Requirements

This topic describes the hardware required for your product installation.

The hardware requirements set by Microsoft for clustered server configurations must be met for systems deploying Workflow with cluster support.

- *Failover* clustering is supported on Workflow 1.5 SP4 and greater.
- SQL Server must be installed in a central location on the domain.



Important:

- All computers in a *failover* cluster must be on a domain.
- *Failover* clustering is not supported on Workflow legacy servers.
- Workflow server clustering provides support against hardware failures only.

Custom services managed by clustering must maintain their state in shared persistent memory (typically a database), and the state must always be in a consistent state because failover can occur at any time.

Related information

[Supported and Validated Cluster Configurations \(on page 103\)](#)

Compatibility

Compatibility and Upgrading with GE Products

If you plan to use one or more other GE products with Workflow, it is important to first determine whether the version of each such product is compatible with the latest version of this application.

Compatible GE Products

Workflow may be used with a number of other GE products. Depending on the product or product version, you may need to run Workflow in a particular architecture mode or take additional measures during set up

to ensure compatibility between the products. If so, the notes in the following table provide the necessary details.

Workflow 2.6 SP2

CIMPLICITY	Historian	iFIX	Open Enterprise	Plant Applications	Operations Hub	Plant Applications Service Provider (PASP) (PASP)	Proficiency Portal	Production Manager
2023	2023	2023	2.2	2023	2023.1	Not Supported	Not Supported	Not Supported
2022	2022	2022		2022	2022.06			
11.1	9.1	6.5		8.2				

Workflow 2.6 SP1

CIMPLICITY	Historian	iFIX	Open Enterprise	Plant Applications	Operations Hub	Plant Applications Service Provider (PASP) (PASP)	Proficiency Portal	Production Manager
8.1	4.5	5.0	2.2	6.3	1.7	Not Supported	Not Supported	Not Supported
8.2	5.0	5.1		6.3 SP1				
9.0	5.5	5.5		6.3 SP2				
9.0 R2	6.0	5.8		7.0				
9.5	7.0	5.9		7.0.1				
10	7.1	6.0						
	7.2	6.1						
	8.0							
	8.1							

Workflow 2.6

CIMPLICITY	Historian	iFIX	Open Enterprise	Plant Applications	Operations Hub	Plant Applications Service Provider (PASP) (PASP)	Proficy Portal	Production Manager
8.1	4.5	5.0	2.2	6.3	1.7	Not Supported	Not Supported	Not Supported
8.2	5.0	5.1		6.3 SP1				
9.0	5.5	5.5		6.3 SP2				
9.0 R2	6.0	5.8		7.0				
9.5	7.0	5.9		7.0.1				
10								

Workflow 2.5 SP4

CIMPLICITY	Historian	iFIX	Open Enterprise	Plant Applications	Operations Hub	Plant Applications Service Provider (PASP) (PASP)	Proficy Portal	Production Manager
8.1	4.5	5.0	2.2	6.3	Not Supported	Not Supported	Not Supported	8.1
8.2	5.0	5.1		6.3 SP1				
9.0	5.5	5.5		6.3 SP2				
9.0 R2	6.0	5.8		7.0				
9.5		5.9		7.0.1				

Workflow 2.5 SP3

CIMPLICITY	Historian	iFIX	Open Enterprise	Plant Applications	Operations Hub	Plant Applications Service Provider (PASP) (PASP)	Proficy Portal	Production Manager
8.1	4.5	5.0	2.2	6.3	Not Supported	Not Supported	Not Supported	8.0
8.2	5.0	5.1		6.3 SP1				
9.0	5.5	5.5		6.3 SP2				
9.0 R2	6.0	5.8		7.0				

Workflow 2.5 SP2

CIMPLICITY	Historian	iFIX	Open Enterprise	Plant Applications	Operations Hub	Plant Applications Service Provider (PASP) (PASP)	Proficy Portal	Production Manager
8.1	4.5	5.0	2.2	6.3	Not Supported	Not Supported	Not Supported	7.5
8.2	5.0	5.1						
9.0	5.5	5.5						
9.0 R2	6.0	5.8						

Workflow 2.5 SP1

CIMPLICITY	Historian	iFIX	Open Enterprise	Plant Applications	Operations Hub	Plant Applications Service Provider (PASP) (PASP)	Proficy Portal	Production Manager
8.1	4.5	5.0	2.2	6.3	Not Supported	Not Supported	Not Supported	Not applicable at time of release
8.2	5.0	5.1						
9.0	5.5	5.5						
9.0 R2	6.0	5.8						

Workflow 2.5

CIMPLICITY	Historian	iFIX	Open Enterprise	Plant Applications	Operations Hub	Plant Applications Service Provider (PASP) (PASP)	Proficy Portal	Production Manager
8.1	4.5	5.0	2.2	6.3	Not Supported	Not Supported	Not Supported	7.0
8.2	5.0	5.1						7.0 SP2
9.0	5.5	5.5						7.0 SP3
9.0 R2	6.0	5.8						7.0 SP4
								7.0 SP5

Workflow 2.2 SP1

CIMPLICITY	Historian	iFIX	Open Enterprise	Plant Applications	Operations Hub	Plant Applications Service Provider (PASP) (PASP)	Proficy Portal	Production Manager
8.1	4.5	5.0	2.2	6.2	Not Supported	2.2	3.5	6.3 SP1
8.2	5.0	5.1		6.3 SP2				
	5.5	5.5		6.3 SP3				
		5.8						

Workflow 2.2

CIMPLICITY	Historian	iFIX	Open Enterprise	Plant Applications	Operations Hub	Plant Applications Service Provider (PASP) (PASP)	Proficy Portal	Production Manager
8.1	4.5	5.0	2.2	6.2	Not Supported	2.2	3.5	6.3
8.2	5.0	5.1						
	5.5	5.5						
		5.8						

Workflow 2.1

CIMPLICITY	Historian	iFIX	Open Enterprise	Plant Applications	Operations Hub	Plant Applications Service Provider (PASP) (PASP)	Proficy Portal	Production Manager
8.1	4.5	5.0	1.6	5.0	Not Supported	1.5	3.5	6.2
8.2	5.0	5.1	6.0					
	5.5	5.5	6.1					
	7.1	6.0						
	7.2	6.1						
	8.0							
	8.1							

**Note:**

The following notes apply to product versions compatible with Workflow 2.5 and Workflow 2.6:

- CIMPLICITY 9.0: End points names must not exceed 32 characters.

The following notes apply to product versions compatible with Workflow 2.2, 2.2 SP1, 2.5:

- CIMPLICITY 8.1 and 8.2: OPC is recommended for exchanging data between Workflow and CIMPLICITY.
- Historian 5.0: Data types added to Historian 5.0 (ihUInt64, ihUInt32, ihUInt16, ihByte, ihBool) are not supported for binding to parameters or local variables in Workflow.

Compatible Upgrade Paths for Workflow 2.6

Workflow 2.6 can be used to upgrade the following versions of Workflow:

- Workflow 2.5 with latest SIM
- Workflow 2.5 SP1, SP2, SP3, SP4 with latest SIM
- Workflow 2.5 Reporting SP1, SP2, SP3 SP4 with latest SIM

- Workflow 2.2 SP1, SP2
- Workflow 2.2 SP1, SP2 Reporting

Compatible Upgrade Paths

The information in this section indicates the current product compatibility when upgrading from Workflow 2.5 to Workflow 2.5 SP1.

Task List

Workflow with:	Workflow client Task List	Supported
	ActiveX Task List	Supported
Vision 6.1 with:	Web Task List	Not Supported
	Workflow Mobile Task List	Not Supported
Vision 6.2 with:	Workflow Mobile Task List	Not Supported
Independent Host with:	Web Task List	Supported
Vision 6.3 with:	HTML Task List	Supported

Service Providers

Historian 4.0, 5.0, 5.5, 6.0	Supported
Plant Applications Service Provider (PASP)	Not Supported
MESCore SP	Supported
Custom Service Provider	Supported



Note:
Service providers that target versions of the .NET framework before version 4.5 must be recompiled targeting .NET 4.5.

Related information

[Architecture Considerations \(on page 54\)](#)

Compatibility Issues with Workflow

There are some specific compatibility issues that have been encountered in Workflow. These compatibility issues are necessary due to certain functionality changes that were made to improve the product.

- Compatibility Issues with Workflow 2.6
 - The same compatibility issues for Workflow 2.5 apply to Workflow 2.6.
- Compatibility Issues with Workflow 2.5
 - [Service Providers \(on page 47\)](#)
 - [Legacy Certificates \(on page 48\)](#)
- Compatibility Issues with Workflow 2.2
 - [Silverlight Forms \(on page 48\)](#)
 - [Custom Application Logging \(on page 48\)](#)

Compatibility Issues with Workflow 2.5

Customer Service Providers

Some DMS methods and fields are now obsolete, and have been replaced, in Workflow 2.5. The deprecated fields should be replaced with the new fully qualified fields. If you do not use these new fields, your code will not compile.



Note:

If you have an existing custom service provider that is both using the deprecated methods and .NET Framework 4.5, you must compile your code again in order to ensure that your service providers function as expected.

Table 1. ModelObject.cs

Obsolete	Replacement
ModelObject.TableName	FullyQualifiedTableName

Table 2. DataSetMap.cs

Obsolete	Replacement
GetTableName()	GetFullyQualifiedTableName()
GetTableNames()	GetFullyQualifiedTableNames()

Table 2. DataSetMap.cs (continued)

Obsolete	Replacement
TableName	FullyQualifiedTableName
HistoryTableName	HistoryFullyQualifiedTableName

**Tip:**

If required, you may choose to use the Table Name that has no schema prefix, such as BaseTableName.

Legacy Certificates

Starting with Workflow 2.5, legacy certificates are no longer supported.

Compatibility Issues with Workflow 2.2**Silverlight 5**

You must upgrade from Silverlight Runtime v4.0 to Silverlight Runtime v5.0, as well as from Silverlight SDK v4.1 to Silverlight SDK v5.0. All forms will automatically be updated.

Custom Application Logging

Custom applications that directly reference Log4Net for logging purposes may stop working with Workflow version 2.2, which installs an updated version of Log4Net. To prevent such compatibility problems going forward, we recommend that custom applications instead be designed to log messages through the ILog interface in `Proficy.Platform.Core.Diagnostics`. If this is not feasible, you must ensure that custom applications reference version 1.2.11 of Log4Net, and be prepared to update the version of Log4Net referenced by custom applications upon upgrade to future releases of Workflow.

Changes to Functionality and Supporting Technology

In planning for upcoming releases of Workflow, please note that a number of changes to functionality and supporting technology are scheduled.

- Support for installing and running Workflow in Legacy (compatibility) mode has been discontinued. It will be possible to upgrade legacy systems to non-legacy systems, but doing so may require updating any service providers that rely on the previous service provider architecture to use the new service provider architecture.

- Support for installing and running the Workflow on Microsoft® Windows XP has been discontinued.
- Support for one-click installation of Workflow will be discontinued in a future release of the software. The client installation should be used when one-click installation is discontinued.
- All Workflow assemblies are now compiled in .NET Framework version 4.5. Custom displays and applications that reference an earlier version of .NET must change to instead reference .NET v4.5 in order to function properly.
- Starting with Workflow 2.5, legacy certificates are no longer supported.

Chapter 2. Get Started

Get Started Guide

Installation Requirements

Key Concepts

Workflow is a unifying architecture for our software products, and a framework to leverage existing applications and functionality. The role of this product is to provide the components and services which, when adopted by a product line, drives common operational behavior and support.

This product gives you a flexible foundation for building applications and interfacing with other software. It contains the tools you need to create applications that can be used to facilitate the management and analysis of activities in your enterprise.

You can customize your environment to:

- host editors for building Workflow applications
- display HMI screens for monitoring plant activities
- display workflow tasks to operators on the plant floor
- accept input into forms that can be created and routed to clients throughout the enterprise

Workflow also provides a configurable event engine that can trigger workflows and other code based on different types of internal and external triggers.

System Health and Program Use

As part of configuring a Workflow system, you indicate which servers should host which product options, and designate servers and product options as either essential or nonessential. These designations in combination with server status determine site health and affect the ability to log in and use this software.

If you designate a server as essential when configuring the system, all associated product options are automatically designated as essential. If you designate any product option as essential, the server that is hosting that option is automatically designated as essential, but the essential status of other product options hosted by the server remains unchanged.

**Note:**

The servers hosting the Core and Workflow product options are designated as essential by default, and this designation cannot be changed.

The designation of servers as either essential or nonessential and the state and accessibility of servers has implications for the health of the system and, therefore, the ability to log in and use the Workflow program, as follows:

- When all servers, both essential and nonessential, are running and reachable, system health is considered *Complete*. In this scenario, the site health indicator in the toolbar is green, users can log in to the program, and all program and product option functionality is available.
- When one or more nonessential servers are not running or are unreachable, system health is considered *Partial*. In this scenario, the site health indicator in the toolbar is yellow, active users remain logged in to the program, and inactive users can log in, but program functionality tied to product options hosted on compromised servers is unavailable.

**Note:**

Specifically, related displays become inaccessible, binding to related external data faults, and related events stop firing. In addition, workflows that consume related data or monitor related events fail (making it a recommended practice to wrap such operations in fault handlers). Upon restoration of server availability, display inaccessibility and data-binding, event-firing, and workflow failure are automatically reversed. However, any work that was being performed using a display when a related server became unavailable may be lost.

- When one or more essential servers that do not host the Core product option are not running or are unreachable, system health is considered *Waiting*. In this scenario, the site health indicator in the toolbar is yellow, active users are logged out of the program, and inactive users cannot log in.

**Note:**

In this case, upon restoration of the essential server(s), users active at the time of interruption are automatically logged back in to the program, and functionality is restored to the level in accordance with the current health of the site.

- When the server that hosts the Core product option (typically, SOAServer) is not running or is unreachable, system health is considered *Unavailable*. In this scenario, the site health indicator in the toolbar is gray, active users are logged out of the program, and inactive users cannot log in.

**Note:**

In this case, upon restoration of the server that hosts the Core product option, users active at the time of interruption are automatically logged back in to the program, and functionality is restored to the level in accordance with the current health of the site.

When system health is either *Complete* or *Partial*, you can use the [System Status \(on page 139\)](#) display to monitor server configuration and status. In these states of site health, you can also [move product options between servers \(on page 136\)](#) to accommodate changing conditions, as necessary.

Related information

[Multiple Servers \(on page 68\)](#)

[Configure product options \(on page 144\)](#)

Regional Settings

The following regional settings are supported.



Note:

Open Enterprise supports the same regional settings as Workflow.

Workflow Client-based Regional Settings

Supported Settings

- Decimal symbol - one character
- Digit grouping symbol
- List separator - one character
- Time style
- Time separator
- Short date style
- Date separator



Note:

The decimal symbol and the digit grouping symbol cannot be the same character. Also, the time separator and the date separator cannot be the same character.

Formatting the Time and Date

Avoid changing the time style or short date style in regional settings to values that are outside of the standard styles provided. Changing these values to non-standard styles may result in improperly formatted times and dates in some parts of Workflow.

This software supports the following short date formats, some of which may not be available in certain language versions of Windows:

- dd/mm/yy or dd/mm/yyyy
- dd/yy/mm or dd/yyyy/mm
- mm/dd/yy or mm/dd/yyyy
- mm/yy/dd or mm/yyyy/dd
- yy/dd/mm or yyyy/dd/mm
- yy/mm/dd or yyyy/mm/dd

Formatting the Regional Language Setting

Avoid changing the language setting when Workflow is running.

Setting the System Default Locale

The selected locale must be set as the system default.

Workflow Web-based Application Settings

In order to capture the date and time settings accurately in Google Chrome and Mozilla[®] FireFox[®], you must change the language settings in these browsers. Changing these settings reflects the date and time that is pertinent to the selected language. For Internet Explorer, your current language settings are automatically applied.

Daylight Saving Time

In Workflow, all dates and times are stored independent of time zones, in UTC format.

Special Keyboard Buttons

Some computer keyboards have special buttons for e-mail launch, Internet launch, search, and other functions. These keyboard buttons may disable certain key macros or allow users to circumvent security measures.

We recommend that you reprogram or disable the software that operates such special buttons. Refer to your computer's documentation for instructions on disabling these buttons.

Log Files

Workflow provides a number of log files that you can use to assist you in troubleshooting various aspects of your system.

On all supported operating systems, the log files are located in: <installdir>\ProgramData
\Proficy\Logs.

**Tip:**

To access the `ProgramData` folder, you must show the hidden folders on your system.

The log files for all installations are also accessible from the program group; that is, **Start > All Programs > Proficy > Proficy Workflow > Logs**.

Compatibility and Upgrading with GE Products

Please refer to the WorkflowIPI PDF for information on upgrading Workflow, compatibility with earlier versions and other GE Products. You can find the WorkflowIPI.pdf in the Documentation folder of the Workflow installation media or in the Proficy Workflow\Help folder after installation. By default this is `C:\Program Files (x86)\Proficy\Proficy Workflow\Help`.

Pre-Installation Consideration

Architecture Considerations

If you are installing an application server or extension server (that is, Workflow or User) on a 64-bit computer, you can configure the server for either 32-bit or 64-bit operation.

When installing for the first time, you indicate your preference at installation time, whereas when upgrading, the installation maintains the existing configuration by default. In both cases, however, components that enable either 32- or 64-bit server operation are installed to your system, and you can subsequently use the Configure Server tool to change between 32- and 64-bit mode, as outlined under **Change architecture mode**.

If you are using a 64-bit operating system and are installing to the default installation location, do not remove (x86) from the default installation directory path. Windows detects that Workflow is a 32-bit process, and defaults to `C:\Program Files (x86)\Proficy...` If you remove (x86) from the default installation location on a 64-bit operating system, your installation will fail.

**Note:**

Client-only installations are restricted to 32-bit operation, regardless of installation location.

Default Installation Locations by Architecture Type

By default, program installation locations by architecture type are as follows:

- 64-bit: `C:\Program Files\Proficy\Proficy Workflow\Program`
- 32-bit: `C:\Program Files (x86)\Proficy\Proficy Workflow\Program`

Determining Current Architecture Mode

Any time after installation, you can consult the **Processes** tab of **Windows Task Manager** to ascertain the architecture mode that the Workflow Server service is currently running under. If operating in 32-bit mode, the `ProficyServer.exe` listing is appended with `*32`.

Implications for Product Options and Service Providers

The **ProficyPackager** and **ProficyInstaller** tools are available in both the 32-bit and 64-bit installation locations, and you can package and install product options from either folder. When you run Workflow application server or Workflow client, the need to update product options is automatically determined depending on where you installed the product options from. For example, if you package and install a product option from the 64-bit folder, and run the 64-bit Workflow application server, the Workflow client (which runs only in 32-bit mode) will automatically invoke the **ProficyUpdater** tool to update the product option in the 32-bit folder.

Product option assemblies that are loaded by the Workflow client (for example, configuration screens and displays) must target either *Any CPU* or *x86*, because Workflow client runs only in 32-bit mode.

Implications for running service providers are as follows:

- Service providers that were built using a target architecture of *Any CPU*, can be run in either 32- or 64-bit mode.
- Service providers that were built to target *x86* (32-bit) architecture can be run only in 32-bit mode.

**Note:**

This is typically the case when the service provider references a C++/CLI library with 32-bit dependencies.

- Service providers that were built to target *x64* (64-bit) architecture can be run only in 64-bit mode.

**Note:**

This is typically the case when the service provider references a C++/CLI library with 64-bit dependencies.

Maintenance of Configuration Settings

After an upgrade installation on a 64-bit computer that was previously running the 32-bit version of Workflow, the server configuration file in the 32-bit location (`C:\Program Files (x86)\Proficy`

`\ProficiencyWorkflow\Program\ProficiencyServer.exe.config`) retains all pre-existing system settings, whereas the configuration file in the 64-bit location (`C:\Program Files\Proficiency\ProficiencyWorkflow\Program\ProficiencyServer.exe.config`) contains new default settings. If you want to switch from 32-bit to 64-bit operation but continue using your previously implemented configuration settings, you should migrate the settings in the 32-bit version of the configuration file to the 64-bit version.

Prerequisites for Windows 2012 R2 Installations

To install the Application Server, Reporting, Extension servers, or ActiveX Task List on Windows[®] Server 2012 R2, the April 2014 update rollup for Windows[®] Server 2012 R2 (KB2919355) must already be applied.



CAUTION:

If this update is missing, the Visual C++ 2015 Redistributable packages installed by the Workflow installer will fail. These packages must be manually uninstalled, and then the necessary Windows updates must be applied before you attempt the Workflow installation again. If you do not uninstall the Visual C++ 2015 Redistributable packages before retrying the Workflow installation, the installation will appear to be successful but the WorkflowServer service will fail to start.



Important:

These recovery steps also apply if you attempt to install Workflow 2.5 SP2 on Windows[®] 7 Professional without SP1 or Windows[®] Server 2008 R2 without SP1.

Configure Windows Authentication for Workflow

You can configure Workflow to use Windows authentication to connect to a remote SQL Server instance.

About this task

To configure Workflow to use Windows authentication, you must use the Windows **Credential Manager** to create a credential specifying the Windows username and password that the Workflow server should use to connect to the database.

The credential on the Workflow server machine must identify the remote SQL Server by IP or fully-qualified domain name (FQDN) and may need to explicitly specify the port depending on the SQL Server configuration.

During the installation of the Workflow server, you will have the opportunity to specify the **hostname** and **port** exactly as it was specified in the credential. Use a comma to separate the hostname and port number per the SQL Server connection string syntax.

Procedure

1. Click **Start > Control Panel > All Control Panel Items > Credential Manager**.
2. Select **Windows Credentials**, and then click **Add a Windows credential**.
The **Credentials** dialog box appears.
3. Enter values into the fields, as follows:
 - Enter the hostname in the **Internet or network address** field. (e.g., `sql2016server:49785`)
 - Enter the username that Workflow should use to connect to the database into the **User Name** field.
 - Enter the password associated with the username you entered into the **Password** field.
 - Click **OK**.
4. In the **Database Configuration** section during the installation of the Workflow server, do the following:
 - Enter the hostname and port number in the **Server** field, just as you did for the credential. (e.g., `sql2016server,49785`)
 - Enter the database name in the **Database** field.
 - Select **Windows Authentication** from the **Authentication** drop-down list.
 - Click **Save**.

Results

Workflow is now configured to use Windows authentication to connect to a SQL Server.

Windows Services for Workflow

There are several Windows services that must be running in order for Workflow to function properly.

Standard Server Installation Services

These services are installed as part of Workflow. They are configured to start automatically on system startup.



Note:

If these services do not start up automatically, you can manually start them, in any order, from the Control Panel's Administrative Tools.

Workflow Server

Supports most of this product's functionality. It must be running at all times.

Proficy STS

Authenticates application users. It must be running at all times.

Proficy Publisher Service

Publishes data to the active directory (Microsoft ADAM/LDS). It must be running if you are using ADAM/LDS.

Workflow Certificate

Generates security certificates.

Support Server Installation Services

The following services are available from other applications:

SQL Server

This service runs on whichever machine the SQL Server is installed on, including the machine the Workflow application server is installed on.

SOAAdam

This service is available if you are using a Microsoft ADAM/LDS instance.

Security Certificates

Security Certificates

Security certificates must be installed on all Workflow application server and client machines in your system.

Security certificates are used to protect your identifiable information and to protect your computers from unsafe software. A certificate is a statement verifying the identity of a person or the security of a website.

During the application installation process, certificates are automatically generated using the Workflow Certificate service. Alternatively, you can select a security certificate for the ProficyPlatform, Proficy STS, and SSL/TLS server services.

You can configure the following types of certificates:

- Self-signed certificates generated during the installation process.
- Existing certificates that you installed and configured for a previous version of the application.

The Workflow client computer must verify and trust the identity of the server before it can securely send a user's login and password credentials and complete the authentication process. To establish this trust, the client must trust the root of the server's certificate. That is, the client must have the certificate of the Certificate Authority (CA) that issued the server certificate in their Trusted Root Certificate Authorities store.

When you generate new certificates or use existing certificates, the following steps will occur.

- Install a trusted root certificate.
- Install a certificate to a Certificate Store (by default, the Personal Certificate Store).
- Install generated self-signed certificates to the proper Certificate Store(s).
- Register the SSL certificates to IP ports assigned to the Workflow application server.
- Provide the option to use existing certificates configured for a previous version of the application.

**Important:**

The SSL/TLS Server Certificate must be unique to each server.

For information on changing or updating security certificates post-installation, see **Modify security certificates**.

Related reference

[Security Certificate Options \(on page 59\)](#)

Related information

[Modify security certificates \(on page 135\)](#)

Security Certificate Options

Server security certificate options differ depending on the type of server you are configuring.

Full Server Certificate Options

The following table describes the type of security certificates available for use on a single full server.

**Note:**

Starting with Workflow 2.5, legacy certificates are no longer supported.

Option	Description
Generate new and unique certificates	<p>This option allows you to automatically generate new self-signed certificates. If you are installing in a multiple server or server failover cluster environment, more configuration is required on those servers.</p>
Import certificates	<p>This option allows you to import certificates that were generated on and exported from a main server. This option is used when installing extension servers in a multiple server or server failover cluster environment and can also be used to install the same certificates on multiple single servers.</p> <p>After selecting this option, in the Certificate File field, click Browse to locate and select the zip file containing the security certificates that you exported.</p>
Use the certificates already installed on this server	<p>Post-installation configuration only: this option is available when you use the Configure Certificates tool. It allows you to use existing certificates that were installed with a previous version of this application.</p> <p>To import certificates, select the Enable certificate import for advanced configuration check box, and then click Import. Click View to view each certificate after it has been imported.</p> <div data-bbox="820 1465 1421 1879" style="border: 1px solid #00a0c0; border-radius: 10px; padding: 10px; background-color: #e6f2f8;"> <p> Note:</p> <ul style="list-style-type: none"> • While importing the “Custom SSL/TLS certificate” the SSL certificate issued by ProficySelfSignedCA is maintained (Added/Removed) from the Windows certificate store and bound to ports required by Workflow. The custom SSL certificate is </div>

Option	Description
	 <p>only imported and bound to ports. The utility does not remove the certificate from Windows certificate store.</p> <ul style="list-style-type: none"> • While importing the custom SSL/TLS Certificate from the utility ConfigureCertificates.exe, if the certificate already exists in the certificate store, then it will need to be deleted manually.

Extension Server Certificate Options

The following table describes the type of security certificates available for use on an extension server; that is, in a multiple server or server failover cluster environment.

Option	Description
Import certificates	<p>This option allows you to import the certificates that are installed on the main server. You must manually export the certificates from the main server to a defined location, and then import the certificates to the extension server. This option will generate an SSL certificate if the main server is using certificates generated by the installation.</p> <p>After selecting this option, in the Certificate File field, click Browse to locate and select the zip file containing the security certificates that you exported.</p>

Client Certificate Options

When you install a remote client, the certificates that you installed on the server (that is, the single server or the main server in a multiple server or server cluster environment) are automatically downloaded and installed on the client.

If the certificates on the server are modified in any way, each client connected to the server will also have to be updated so that the certificates match those on the server.

Related information

[Security Certificates \(on page 58\)](#)

[Modify security certificates \(on page 135\)](#)

Install Certificates on older Operating Systems

You can create certificates on a Windows 10 or Windows Server 2016 machine and transfer them to a machine with an older operating system.

About this task

If you would like to use valid, self-signed certificates for a Workflow installation on an operating system other than Windows 10 or Windows Server 2016, you can do the following:

1. Manually generate the certificates using **Powershell New-SelfSignedCertificate** on a Windows 10 or Windows Server machine.
2. Export the certificates using **MMC**.
3. Import the certificates to the selected Workflow server machine using Workflow **ConfigureCertificates**.
4. Import the SSL certificate using Workflow **ConfigureCertificates**.

Procedure

1. Manually generate the certificates on a Windows 10 or Windows Server 2016 machine.
 - a. Use the following **New-SelfsignedCertificate** parameters to generate the certificates:

Certificate Name	New-SelfSignedCertificate command parameters
ProficySelfSignedCA	New-SelfSignedCertificate -CertStoreLocation "cert:LocalMachine\My" -DnsName "ProficySelfSignedCA" -FriendlyName "ProficySelfSignedCA" -HashAlgorithm SHA256 -KeyExportPolicy Exportable -KeyLength 2048 -KeySpec KeyExchange -KeyUsage CertSign, CRLSign, DigitalSignature -KeyUsageProperty All -NotAfter \$(date-

Certificate Name	New-SelfSignedCertificate command parameters
	time]::now.AddYears(3)) -Subject "CN=ProficySelfSignedCA"
ProficySTS	<pre>\$proficyCACert = Get-ChildItem -Path cert:\LocalMachine\My ?{\$_.Subject -eq "CN=ProficySelfSignedCA"} New-SelfSignedCertificate -CertStoreLocation "cert:\LocalMachine\My" -DnsName "ProficySTS" -FriendlyName "ProficySTS" -HashAlgorithm SHA256 -KeyExportPolicy Exportable -KeyLength 2048 -KeySpec KeyExchange -NotAfter \$('[datetime]::now.AddYears(3)') -Signer \$proficyCACert -Subject "ProficySTS"</pre>
ProficyPlatform	<pre>\$proficyCACert = Get-ChildItem -Path cert:\LocalMachine\My ?{\$_.Subject -eq "CN=ProficySelfSignedCA"} New-SelfSignedCertificate -CertStoreLocation "cert:\LocalMachine\My" -DnsName "ProficyPlatform" -FriendlyName "ProficyPlatform" -HashAlgorithm SHA256 -KeyExportPolicy Exportable -KeyLength 2048 -KeySpec KeyExchange -NotAfter \$('[datetime]::now.AddYears(3)') -Signer \$proficyCACert -Subject "ProficyPlatform"</pre>
SSL certificate	<pre>\$proficyCACert = Get-ChildItem -Path cert:\LocalMachine\My ?{\$_.Subject -eq "CN=ProficySelfSignedCA"} New-SelfSignedCertificate -CertStoreLocation "cert:\LocalMachine\My" -DnsName "localhost","<ipAddress>" -FriendlyName "localhost" -HashAlgorithm SHA256 -KeyExportPolicy Exportable -KeyLength 2048 -KeySpec KeyExchange -NotAfter \$('[date-</pre>

Certificate Name	New-SelfSignedCertificate command parameters
	<pre>time]::now.AddYears(3)) -Signer \$proficy- CACert -Subject "<hostname>"</pre> <p>Where <ipAddress> and <hostname> are the IP Address and the Hostname of the Workflow application server machine.</p>

- b. Run **Powershell** as an administrator to create the certificates.
2. Export the certificates from the **Local Computer > Personal** store, using the Microsoft Management Console (MMC) certificate snap-in.
 - a. For each certificate, select **export the private key**.
 - b. On the **Export File Format** dialog, select **Personal Information Exchange - PKCS #12 (.PFX)** format.
 - c. Uncheck the **Include all certificates in the certification path if possible** check box.
 - d. Check the **Export all extended properties** check box.
 - e. Export all of the certificates to the same directory, naming each **.pfx** file with the certificate name (Issued To) of the certificate it contains. Use the same password for each certificate.
 - f. Combine the certificates into a **.zip** file and copy the .zip file to the Workflow server machine.
 - g. Unzip the file for use.

**Note:**

- If you leave the **Include all certificates in the certification path if possible** check box selected, the **ProficySelfSignedCA** cert is re-imported into the **Local Machine > Personal** store when you import the **SSL** certificate in a later step. You will then have to remove it.
- Exporting using the same password and naming each file with the **Issued To** name is mandatory for importing on the Workflow application server using **ConfigureCertificates**.

3. Import the certificates using Workflow **ConfigureCertificates**.
 - a. On the selected Workflow server, run:

```
C:\Program Files (x86)\Proficy\Proficy Workflow\Program\ConfigureCertificates.exe
```
 - b. Select the **Import Certificates** option.

- c. Select the **.zip** file you copied for importing (not the individual .pfx files).
The three Proficy certificates are imported from the .zip files into the correct locations.
- d. Click **Save**.
Once saved, an SSL certificate is generated and bound to the Workflow ports. This SSL certificate must be removed before proceeding to the next step.
- e. Using MMC, remove the **SSL** certificate from the **LocalMachine > Personal** store. You will replace this certificate in the next step.

4. Import the **SSL** certificate using Workflow **ConfigureCertificates**.

- a. On the selected Workflow server, run:

```
C:\Program Files (x86)\Proficy\Proficy Workflow\Program\ConfigureCertificates.exe
```
- b. Select **Use the certificates already installed on this server** option.
- c. Select the **Enable certificate import for advanced configuration** check box.
- d. Click the ellipses (...) in the **Import** column for the **SSL/TLS Server Certificate**, and browse to the folder containing the exported certificates.
- e. Select the **SSL** certificate. This is the certificate with the **Issued To** equal to the Workflow application server hostname.



Note:

You may need to change the file browser selection from **.cer** to **All Files** to see **.pfx** file.

- f. Follow the prompts to import the **SSL/TLS Server Certificate**.
- g. Click **Save** to bind the Workflow ports to this certificate.

Related reference

[Security Certificate Options \(on page 59\)](#)

Related information

[Use self-signed certificates for the web server \(on page 66\)](#)

[Security Certificates \(on page 58\)](#)

[Modify security certificates \(on page 135\)](#)

Use self-signed certificates for the web server

If you are using self-signed certificates, you must install them on client machines in order to access the web server. Download the certificate to your client machine by accessing `http://yourservername:8008/Certs/Help`.

Procedure

1. From the root certificate link, save the certificate file as `Root.cer`.
2. Open a command prompt window.
3. At the command prompt, enter **mmc**.
4. From the **File** menu, click **Add/Remove Snap-in**.
5. Select **Certificates**, and then click **Add**.
The **Certificates snap-in** dialog box appears.
6. Select **My user account**, click **Finish**, and then click **OK**.
7. Expand **Certificates - Current User > Trusted Root Certificate Authorities > Certificates**.
8. Right-click **Certificates**, select **All Tasks**, and then click **Import**.
The **Certificate Import Wizard** appears.
9. To start, click **Next**, and then in the **File Name** field, enter `Root.cer`.
10. Click **Next**.
11. Select Place all certificates in the following store.
12. Click **Browse**.
13. Select **Trusted Root Certificate Authorities**.
14. Click **Next**, and then click **Finish**.
15. Close all open windows.
16. Restart your browser, and then connect to the server.



Note:

For Firefox users, perform the following steps:

- Navigate to your web server; for example, `https://yourservername`.

The **This Connection is Untrusted** page appears.

- Expand the section, **I Understand the Risks**.
- Click **Add Exception**.
- Click **Confirm Security Exception**.

**Note:**

If this page does not refresh, clear the check box **Permanently store this exception**, and then try again.

- You can now access the server from this browser.

Related reference

[Security Certificate Options \(on page 59\)](#)

Firewall Ports

During server installations, the installation setup detects whether there is a firewall on the computer. You must either disable the firewall or configure it to allow communication with remote clients.

The following is a list of the default inbound TCP ports you must open on the server computer to allow communication:

- 8447
- 8012
- 8111
- 8112
- 8020
- 8201
- 8202
- 8203
- 8204

**Important:**

As of Workflow version 2.6, the default port for REST services hosted by Workflow is changed from 8444 to 8447. The REST service port is NOT changed during upgrade of existing installations and remains 8444.

GE Web HMI and GE Workflow do not work together if installed on the same machine as both applications require the use of port 8444. You must change the port number in your Workflow server information in Web HMI. If no port information is provided, the default is used. For information about resolution of port conflicts, see [Resolve Port conflicts between prior Workflow installations and Web HMI \(on page 329\)](#).

For existing Workflow installations prior to 2.6, Workflow and Web HMI ports are both set to 8444, making applications installed on the same machine incompatible.

Database Backup and Restore

Workflow utilizes SQL Server to store information. In order to ensure that the integrity of the information is maintained, it is important to back up your database on a regular basis.

In addition to backing up and restoring your SQL Server database, maintenance plans can assist in keeping your systems up and running. Refer to your SQL Server manual or the Microsoft web site for additional information on backing up, restoring, and maintaining your SQL Server database.

Multiple Servers

Workflow provides the ability to install across multiple servers, each of which hosts a set of services. A multi-server environment is transparent and appears to be a single server to remote clients.

Operational Overview

In a multi-server Workflow site, each installed product option may be hosted by a different server, and each server must be able to determine where each product option resides. To accomplish this, the servers refer to a shared configuration data model. Client programs, such as forms or custom clients, can connect to any server and be redirected to the server that is hosting the service(s) required for the product option in question. Because client programs communicate with product options using service methods and events, neither the user nor the client program is aware of which server is hosting the product option.

Multi-server Considerations

Configuring your system with multiple servers, whether on a single machine or on multiple machines, provides various benefits.

- If you want to run a 32-bit service provider but your system is configured as 64-bit, you can configure a user server to host the 32-bit service provider.
- If you want to run a non-essential service provider but doing so may cause your system to crash, you can configure a user server to host the service provider, thus isolating it from the SOA Server and preventing a system failure.
- If you want to conduct performance monitoring, you can configure a user server and divide the service providers into groups to determine which service provider(s) is causing memory leaks.

- A user server that hosts custom service providers can be restarted more quickly than a single SOA Server that hosts all of your service providers and product options. When new service providers are added to the user server, only the user server needs to be restarted, reducing down time.
- 32-bit machines allow only 3 GB of memory per process. Running multiple servers provides a workaround to this memory constraint.

Configuring your multi-server system on a single machine provides various benefits over configuring a multi-machine system.

- Configuring multiple servers on a single machine is faster and easier, especially in terms of security certificates. After the security certificates have been configured for the main SOA Server, those certificates apply to all servers on that machine. If you configure servers on multiple machines, you must configure security certificate on each machine.
- Communication times between servers and clients is faster on a single machine setup.
- Configuring multi-server failover clustering is faster and easier.

Configuration Overview

In addition to the primary SOA Server instance, which by default hosts both the Core and Workflow product options, you can configure extension servers. All of these servers can be configured on either separate machines or on a single machine. These servers can take the form of a Workflow server, to host the Workflow product option, and/or one or more User servers, to host product options other than the Core and Workflow options. Upon installing a Workflow server, the Workflow product option and related services automatically move from the SOA Server instance to the Workflow server instance. Likewise, upon installing a User server, product options other than the Core and Workflow options automatically move to the User server instance.

**Note:**

Workflow and User servers should not be installed at multiple sites across a WAN.

After installing extension servers, you can use the Configure Site tool **Configure Site** tool to move product options and related services among server instances.

**Note:**

The Core product option and services must reside on the SOA Server instance.

Beginning with Workflow 2.1, you can also designate whether servers and/or product options are essential to your use of the program, as well as disable any product options designated as nonessential.



Note:

The Core and Workflow product options and services and the servers that host them cannot be designated as nonessential.

The designation of servers and production options as either essential or nonessential and the state and accessibility of servers has implications for the health of the site and, therefore, the ability to log in and use the Workflow program. For more information, see **System Health and Program Use**.

For more information about implementing a multiple-server installation of Workflow, see **Architecture Considerations**, **Multiple Server Installation on Multiple Machines**, and **Server Clustering and Failover**.

Related information

[Architecture Considerations \(on page 54\)](#)

[Configure multiple servers on a single machine \(on page 87\)](#)

[Multiple Server Installation on Multiple Machines \(on page 90\)](#)

[Server Clustering and Failover \(on page 70\)](#)

[Modify site configuration \(on page 136\)](#)

[System Health and Program Use \(on page 50\)](#)

Server Clustering and Failover

Workflow provides the ability to implement Microsoft® Windows Clustering, which allows you to configure your server environment to be fault tolerant.

Workflow allows you to configure a *failover* cluster to maintain a consistent image of the cluster on all nodes. It also allows nodes to transfer resource ownership on demand.

A cluster is a group of independent computer systems working together as a unified computer resource. A cluster provides a single name for clients to use, a single administrative interface, and guarantees that data is consistent across nodes.

Workflow works with the Microsoft Failover Cluster Manager to ensure high availability of the Workflow application server. If the primary server node in the cluster experiences difficulties, Workflow is automatically started on another node to take over (a process known as failover). Server high availability is managed through the Microsoft Cluster Manager.

The following information will guide you through the deployment of Workflow in a clustered environment.

1. Overviews of the **Supported and Validated Cluster Configurations**.
2. Instructions for using the **Microsoft Failover Cluster Manager for Windows Servers** to create a cluster.
3. Instructions for installing the **Microsoft Failover Cluster Manager**.
4. Instructions for configuring the **Microsoft Failover Cluster Manager**.
5. Instructions for configuring the **Primary Cluster Server**.
6. Instructions for **Export Server Security Certificates for an Extension Server** from the primary cluster server in preparation for installing them on the failover server.
7. Instructions for configuring the **Failover Cluster Server**.

Assumptions

It is assumed that the reader has a thorough knowledge of the following information:

- Microsoft[®] Failover Clustering
- Microsoft Windows server environments

Additionally, it is assumed that the instructions and guidelines provided by Microsoft for deploying server clusters using Microsoft Windows servers has been followed. For information about server clusters, visit the Microsoft TechNet web site.

SQL Server Clustering

Microsoft also supports clustering. For complete information about installing, configuring, and maintaining SQL Server *failover* clustering, see the following topics on the Microsoft Development Network web site: SQL Server Failover Cluster Installation; or, Getting Started with SQL Server 2008 R2 Failover Clustering.

You can also visit the Microsoft Development Network web site to access information about the high availability features supported by the various editions of SQL Server.

Related reference

[Server Clustering and Failover Software Requirements \(on page 32\)](#)

[Server Clustering and Failover Hardware Requirements \(on page 39\)](#)

Related information

[Install a single application server and client \(on page 79\)](#)

[Configure a server instance \(on page 143\)](#)

[Microsoft Failover Cluster Manager for Windows Servers \(on page 108\)](#)

[Supported and Validated Cluster Configurations \(on page 103\)](#)

[Install Microsoft Failover Cluster Manager \(on page 109\)](#)

[Configure the Microsoft Failover Cluster Manager \(on page 109\)](#)

[Configure the primary cluster server \(on page 112\)](#)

[Export server security certificates for an extension server \(on page 91\)](#)

[Configure the failover cluster server \(on page 113\)](#)

Enable Connection between Workflow and SQL Server

In order for Workflow to connect to any edition of Microsoft® SQL Server 2012 or later, you must configure SQL Server to enable the *db_owner* role on the SOADB database for the applicable login account.

If you are using Windows authentication for Workflow, the default login account is *NT AUTHORITY \SYSTEM*, which is predefined in SQL Server.

For increased security when using Windows authentication, you can modify the properties for the Workflow Server service after installing Workflow, to define a custom login name and credentials. (For more information, see Microsoft® Windows help.) Another alternative is to select SQL Server authentication when installing Workflow. In either of these scenarios, you must add the user-specified login account to SQL Server, and then enable the *db_owner* role for that account.

For information on adding a login account to SQL Server, mapping a login to a database, and setting database roles, see the Microsoft® SQL Server help.

License Management

The GE License Client provides a single, easy-to-use tool to both view and manage online software licensing.

About this task

To access the online help for the GE License Client, perform the following step.

Procedure

From the desktop Start Menu, select **General Electric > License Help**.

The License Client online help opens.

**Note:**

For more information regarding licensing, see <http://support.ge-ip.com/licensing>

Performance Counters

Performance Counters

Windows performance counters are exposed as objects with counters. The counters are grouped into two categories: DTL and Events.

Installation

When the application server is installed, the performance counters are installed automatically and uninstalled when the server is uninstalled.

DTL Counters

You can use DTL counters in Workflow.

Counter Name	Description
DTL Pending Packets Acknowledged	Shows how many non-specific packet acknowledgements are in the queue. If this number starts increasing, it indicates that DTL is having difficulties processing.
DTL Received Packets Queued	Monitors the queue from the RxThread to PacketProcessThread. If you are generating many events with multiple clients, this counter should increment.
DTL Received Packets Dropped	DTL limits the size of the queue from the RxThread to the PacketProcessThread to a maximum of 1,000 (ReceiveQueueLimit) packets. If the RxThread gets a packet when the queue is full, the pack-

Counter Name	Description	
		et is dropped, and this counter increments.
DTL Send Packets Queued		Monitors the queue of packets to be sent, which is added to by the BuildPacketThread and emptied by the TxThread. Normally, this queue is not busy. It indicates that something is generating packets faster than they are being transmitted. If your network connection is slow, or is very busy and generating a high volume of events, this counter may increase.
DTL Record Rate Changed		Reports the number of change records per second coming into the DTL subsystem (typically from an event subsystem), which is the event producer side of the multi-server system. When events are generated, this rate counter moves.
DTL Change Records Queued		Monitors the queue of change records coming from the Events subsystem (and possibly others) that are emptied by the BuildPacketThread. Because the BuildPacketThread pauses to try to get multiple changes per packet, this counter should change when there are a high volume of events.
DTL Change Records Dropped		The queue of change records is limited to 1,000. If more change

Counter Name	Description	
		<p>events come in than can fit in the queue, DTL starts dropping change records, and this counter increases. Typically, this counter does not move, but if the events are generated fast enough, it may move.</p>

Events Counters

You can use events counters in Workflow.

Counter Name	Description	
Events Queued Messages to Subscribers		<p>Monitors the queue in the events subsystem for the subscribers. Typically, a high volume of events cause this rate counter to increment. If there are N subscribers to a given event, there will be N Events Queued Messages to Subscribers for each event generated.</p>
Events DTL Subscriptions		<p>Records the number of DTL points being subscribed to.</p> <p>The event subsystem re-uses DTL subscriptions so that multiple event clients requesting the same DTL point create only one subscription. Thus, if there are three WorkflowClients subscribing to the WorkflowTaskInstanceEvent, there should be one Events DTL Subscription and three Events Remote Subscriptions.</p>

Counter Name	Description	
Events Remote Subscriptions	<p data-bbox="1024 275 1390 443">Records the number of subscribers to events on another server (such as those requiring DTL subscription).</p> <p data-bbox="1024 485 1414 961">The event subsystem re-uses DTL subscriptions so that multiple event clients requesting the same DTL point create only one subscription. Thus, if there are three WorkflowClients subscribing to the WorkflowTask-InstanceEvent, there should be one Events DTL Subscription and three Events Remote Subscriptions.</p>	
Events DTL Change Rate	<p data-bbox="1024 999 1419 1612">Reports the number of events per second being fed to the event subsystem from DTL, which is the subscriber side of the multi-server system. When events are generated, this rate counter changes. The DTL Change Record Rate and the Events DTL Change Rate represent the input side and output side, respectively, of the DTL link for events and, in a stable system, should be the same value (with some room for movement).</p>	
Events Messages to Subscribers Rate	<p data-bbox="1024 1648 1398 1852">Reports the number of events per second being sent to subscribers. In a stable system, if there are N subscribers to a given event, the Events Messages</p>	

Counter Name	Description	
		to Subscribers Rate will be <i>N</i> times the Events DTL Change Rate (assuming only that event is being generated). If the Events Messages to Subscribers Rate is less than <i>N</i> times the Events DTL Change Rate, the Events Queued Messages to Subscribers should increase.

Log performance counters

Using this procedure, you can write (increment/decrement) to the performance counter.

Procedure

1. To add performance counting to your system, select the `ProficyServer.exe.config` file from the following folder: `<installdir>\Proficy\Proficy Workflow\Program`.
2. By default, the performance counter key is false. To start logging performance, set it to true in the following code.

```
<appSettings>
  <add key="EnablePerformanceCounters" value="false"/>
</appSettings>
```



Note:

When **EnablePerformanceCounters** is false, then the logging counters do not increment or decrement. If the value of the key is changed (from either true to false or false to true) after the server is started, the server must be re-started for this change to take effect.

Add or remove performance counters

Using this procedure, you can manually add performance counters to your system, or remove them. You must have administrative privileges in order to perform this action.

Procedure

1. From the following location, you can add or remove performance counters: `<installfolder>/ProficyServer.exe`.
2. In `ProficyServer.exe`, enter the applicable command line.

To...	Enter...
Add counters	<code><installfolder>ProficyServer[exe] / installperformancounters</code>
Remove counters	<code><installfolder>ProficyServer[exe] / uninstallperformancounters</code>

**Important:**

If the counters are added/deleted (installed/uninstalled) after the server is started, then the server and the performance monitoring tool (perfmon) must be re-started for this change to take effect.

Related information

[View performance counters \(on page 78\)](#)

View performance counters

You can view performance counters using the Microsoft Windows **Reliability and Performance Monitor** application.

Procedure

1. Click **Start > Run**.
2. In the **Open** field, enter `perfmon`, and then click **OK**.
3. From **Monitoring Tools**, select **Performance Monitor**.
A graph region appears.
4. Right click the graph region, and from the menu, select **Add Counters**.
The following categories appear: DTL and Events.
5. Expand these categories to view more information.

Related reference

[DTL Counters \(on page 73\)](#)

[Events Counters \(on page 75\)](#)

Related information

[Add or remove performance counters \(on page 77\)](#)

Installation Procedures

Install a single application server and client

The following information guides you through the full installation process of the application server and client.

About this task



Note:

During this installation process, you are required to create an Administrator user. This user is automatically assigned all permissions, which will allow you to create and define your user and security permissions.

Procedure

1. From the application splash screen, click **Install Client and Application Server**.



Note:

If the splash screen does not appear, run `InstallFrontEnd.exe` on the root directory of the installation folder.

The setup wizard appears.

2. Click **Application Server**.

The **Application Server Type** page appears.

3. Select **Core server**, and then click **Next**.

The **License Agreement** page appears.

4. Review the license agreement, and then click **I Agree**.

The **Installation Folders and Architecture** page appears.

5. Proceed as applicable based on the architecture of your computer.

If the architecture is...	Then do this...
32-bit	<ol style="list-style-type: none"> a. Accept the default destination folder or browse for a new location, and then click Next. <p>The AD LDS Integration page appears.</p>

If the architecture is...	Then do this...
<p>64-bit</p>	<p>a. Accept the default Destination Folder (32-bit), or browse for a new location.</p> <div data-bbox="938 386 1422 697" style="border: 1px solid #ffc107; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Important: This is the installation location for 32-bit components. Do not install 32-bit components to the 64-bit installation directory; otherwise, the installation will fail.</p> </div> <p>b. Accept the default Destination Folder (64-bit), or browse for a new location.</p> <p>c. Select either 32-bit or 64-bit depending on the architecture in which you want the server to operate.</p> <div data-bbox="938 978 1422 1335" style="border: 1px solid #17a2b8; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Tip: The default is 64-bit, but you may also select 32-bit. If your environment changes after installation, you can use the Configure Server tool to change the architecture.</p> </div> <p>d. Click Next.</p> <p>The AD LDS Integration page appears.</p>

6. To integrate Active Directory Lightweight Directory Services (AD LDS), do one of the following:

If AD LDS is...	Then...
<p>NOT installed</p>	<p>You are given information regarding its use. If it is not required, then click Next.</p>
<p>Installed in a test and/or development environment</p>	<p>Select the Do not use Microsoft Directory Services check box, and then click Next.</p>

If AD LDS is...	Then...
Installed in a production environment	Click Next .

7. If you are using AD LDS, perform the following procedure:
 - a. Clear the **Do not use AD LDS** check box.
 - b. In the **Directory Instance** section, in the **Name** field, enter an instance name for the ADAM directory.
 - c. In the **Port (LDAP)** and **Port (SSL)** fields, enter valid port numbers.
 - d. In the **User Credentials** section, in the **Name** field, enter the name of a user that is a member of a local computer's Administrators Group.
 - e. In the **Domain** field, enter the name of your company's domain.
 - f. In the **Password** field, enter the password for the user you entered in the **Name** field.
8. Click **Next**.
The **Database Configuration** page appears.
9. Specify the SQL Server database settings for this application based on the configuration of your environment.

**Note:**

To create a backup copy of the database that you choose, select **Back up your existing database**.

To configure SQL Server settings...	Then...
Automatically	<ol style="list-style-type: none"> a. Select the Use a local database with Windows Authentication check box. <div data-bbox="938 1381 1422 1780" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin: 10px 0;">  <p>Note: To use this option, the following SQL Server conditions apply:</p> <ul style="list-style-type: none"> ▪ Installed locally, AND ▪ Designated as default Local Host, AND ▪ Uses Windows Authentication </div> b. Proceed to step 10.

To configure SQL Server settings...	Then...
Manually	a. Clear the Use a local database with Windows Authentication check box. b. Proceed to step 9.

10. Specify your SQL Server settings, as follows:

a. In the **Server** field, enter or select the name of the SQL Server that you want to connect to.



Note:

If the SQL Server is installed locally with a default instance you may enter
localhost.

b. In the **Database** field, enter the name of your SQL Server database, or click the drop-down arrow to search for all databases located on the specified server.



Note:

If the specified database does not exist, it will be created for you.

c. From the **Authentication** list, select the type of authentication that you want to use, and then proceed based on your selection.

- If you select **Windows Authentication**, proceed to step 10.
- If you select **SQL Server Authentication**, proceed to step d.

d. Enter the **User Name** and **Change Password** that are configured for SQL Server authentication, and then proceed to step 10.

Authentication:

- Password character restrictions: <, >, &, '.
- The account used for SQL Authentication must have access to create new databases.
- To install the application server and related SIMs, the SQL Server Management Studio server role, *db_owner*, is required.
- For regular database use during run time, the SQL server login account requires *db_owner* and *dbcreator* privileges. After adding these roles in SQL Server Management Studio, run the Configure Database tool to update log on credentials. For more information, see the **Modify your Workflow application server database** section in the Workflow help.

11. Click **Next**.

The **Security** page appears.

12. Specify the administrator's authentication credentials.

- a. In the **Name** field, accept the default name or enter a new name.
- b. In the **Password** field, enter a password.



Note:

If password complexity is enabled and configured, the Administrator password must follow the same rules as defined for user passwords.

- c. In the **Confirm Password** field, enter the password again.

13. **Optional:** You can specify one or more advanced authentication settings. If you do not want to use this feature, clear all check boxes, and then click **Next**.

- a. To use GE SSO authentication:
 - i. Select the **Use SSO (Single Sign On)** check box.



Note:

Selecting this check box displays the SSO Authentication option when a user logs on,

- ii. Select an identify provider option that complies with the level of restrictions and security required for your application.



Note:

The **Production Identity Provider** option provides greater security for your application.

- b. Select the **Allow Password Change** check box if you want to allow users to change their login password at any time.

Selecting this check box displays the Change Password option when a user logs on.
- c. Select the **Enforce User Lockout** check box, and then set the lockout threshold, duration, and timeframe values. This check box is selected by default.



Tip:

The Administrator account is also restricted by the lockout settings.

- d. Select the **Enforce Password Complex Rules** check box, then select Simple, Normal, or Advanced to determine the level of complexity for your passwords.
On new installations, this check box is selected by default. When upgrading versions, this check box is cleared by default.
- e. Select Web HMI Access and enter the Web HMI Host name if you wish to access the **Workflow Task List** from your Web HMI installation.

14. Click **Next**.

The **Security Certificates** page appears.

15. Select the appropriate certificate option. For full descriptions of the certificate options, see **Security Certificate Options**.

16. Click **Next**.

If the computer you are installing on has an enabled firewall, the **Firewall Settings** page appears.

17. If required, make note of the port numbers to open in the firewall, and then click **Next**.



Important:

If your system uses a firewall, you must follow the instructions on the **Firewall Setting** page and open ports 8447, 8012, 8111, 8112, 8020, 8201, 8202, 8203, and 8204 to incoming TCP traffic. For information on opening a port in your firewall, see online help for your Windows operating system.

The **Installation Confirmation** page appears.



Tip:

At any time during the configuration process, you can click **Back** to return to a previous page to change your settings.

18. Click **Install**.

The **Installation** page appears, displaying the status of each installation step.



Note:

In certain situations, a system restart is required after installation. If this is the case, a corresponding message is displayed among the listed status messages.

19. When the installation is complete, click **Exit**.

**Note:**

If the installation fails, or you want to change a configuration setting after the installation completes, you can open the appropriate standalone component to configure the required information. For more information, see **Post Installation Configuration**.

Results

The Web Task List is automatically installed with the Workflow application server installation. Users can access the WebTask List using a web browser. For more information, see **Connect to the Web Task List**.

Install a remote client

This information guides you through the process of installing a remote client on a different machine than the application server.

Before you begin

Install a single application server and client.

**Note:**

You must install the application server before you install a client because you require an available server to connect to.

Prerequisites

[Install a single application server and client \(on page 79\)](#)

Procedure

1. From the application splash screen, click **Install Workflow Client and Application Server**.

**Note:**

If the splash screen does not appear, run `InstallFrontEnd.exe` on the root directory of the installation folder.

The setup wizard appears.

2. Click **Client Only**.
The **License Agreement** page appears.
3. Review the license agreement, and then click **I Agree**.
The **Installation Folder** page appears.

4. Accept the default destination folder or browse for a new location, and then click **Next**.

The **Application Server** page appears.

5. Specify the name of the application server that this client will connect to.

- a. In the **Computer Name** field, enter the name of the computer where the server you want to connect to is installed.

- If you are configuring a full server or in a multiple server environment, enter the hostname of the computer on which the Workflow server is installed.
- If you are configuring a server cluster environment, enter the cluster name used when you set up your cluster.



Note:

You can also use the IP address for the server if the server uses a static IP address.

- b. In the **Instance Name** field, enter the name of the server instance you want to connect to.

- c. In the **HTTP Port** field, enter the port number required to allow communication with the server.

- d. Click **Next**.

The **Security Certificates** page appears.

6. The security certificates are automatically downloaded from the Workflow server to which you are connecting.

7. Click **Next**.

The **Installation Confirmation** page appears.



Tip:

At any time during the configuration process, you can click **Back** to return to a previous page to change your settings.

8. Click **Install**.

The **Installation** page appears, displaying the status of each installation step.



Note:

In certain situations, a system restart is required after installation. If this is the case, a corresponding message is displayed among the listed status messages.

9. When the installation is complete, click **Exit**.
10. Launch the Workflow client.
A message appears stating: Updates are available. Click OK to start Workflow Update.
11. Click **OK**, and then, on the **Workflow Update** page, click **Install Updates**.
12. When the updates have finished installing, click **Restart Client**.

Configure multiple servers on a single machine

After installing application server, you can configure a multiple server environment onto a single machine. This procedure demonstrates how to create two additional server instances; however, you can create as many instances as you require.

Before you begin

Install a single application server and client.



Important:

- You must exit all instances of Workflow that are located on the server that you are working on.
- You must stop all running workflows on your system before creating a new server instance and corresponding services.
- For each server instance, use a unique server name (see step 2b) and port number (see step 2f).

About this task

You can configure multiple servers on a single machine for two main reasons.

- If your SOAServer must run on a 32-bit system and is approaching 3 GB of memory usage, then you can move workflows or service providers to a separate server instance to reduce memory usage on the SOAServer.
- If custom service providers are unstable and you must restart SOAServer frequently, you can move customer service providers to separate server instances.

Procedure

1. Open the **Services** window to stop all services.
 - a. From the **Start** menu, go to **Control Panel > Administrative Tools > Services**.
 - b. In the **Name** column, click **Stop the service** for each of the following services: Proficy Server, Proficy STS, Proficy Publisher Service.

2. Create a new server instance using the **Configure Server** tool.

- a. On the server machine, On the server machine, click **Start > All Programs > General Electric > Workflow > Configuration > Configure Server**.
- b. In the **Name** field, enter a name for the new server instance, such as `WorkflowServer`.
- c. In the **Description** field, enter a description for the server instance.
- d. Click the **Advanced Configuration** arrow.
- e. Select **64 bit** to run the server using that architecture.



Note:

We recommend that you use a 64-bit system unless your service provider requires 32-bit.

f. In the **P2P Configuration** section:

- In the **Port** field, enter the port number for the server instance. This value must be unique if multiple server instances are run on the same machine.



Note:

For the **IP Address** field: Peer servers will contact this server instance at the specified IP address. If the specified address is `255.255.255.255`, then the address will be determined at server start as the first IPv4 address found in the DNS records for the hosting machine's host name. If the address is incorrect, then an appropriate address should be specified here.

g. Click **Save**.

The following service has now been created: Proficy Server - Workflow Server.

h. Click **Back**.

- i. In the **Name** field, enter a new name for `SOAServer`, such as `UserServer32`.
- j. In the **Description** field, enter a description for the server instance.
- k. Click the **Advanced Configuration** arrow.
- l. Select **32 bit** to run the server using that architecture.

- m. In the **Port** field, enter the port number for the server instance. This value must be unique if multiple server instances are run on the same machine.
- n. Click **Save**.



Important:

After saving the entered data, port numbers are validated to ensure that the same one is not used more than once. To find a new port number, enter a number that is close to the existing one. If a port is already in use, an error will occur. You are then required to return to the entry fields and enter all data again.

The following service has now been created: Proficy Server - UserServer32.

3. Move service providers to the new server instances using the **Configure Site** tool.
 - a. On the server computer, click **Start > All Programs > General Electric > Workflow > Configuration > Configure Site**.
The **Configure Site** tool appears.
 - b. In the **Application Server Instances and Enabled Product Options** pane, select the service provider from `SOAServer` that you want to transfer to the new instances, and then drag and drop it below the new host servers, `WorkflowServer` and `UserServer32`. Repeat this process until all service providers have been moved.
 - c. Click **Save**.
4. Open the **Services** window to start each service again.
 - a. In the **Name** column, click **Start the service** to start each of the following services again: Proficy Server, Proficy STS, Proficy Publisher Service, Proficy Server - UserServer32, Proficy Server - WorkflowServer.
5. Launch the Workflow client using either the **Start** menu or desktop icon.

Revert multiple servers on a single machine back to a single server

After configuring multiple servers on a single machine, you can return to your original single server environment.

Before you begin



Important:

You must exit all instances of Workflow that are located on the server that you are working on.

Procedure

1. On the server computer, click **Start > All Programs > General Electric > Workflow > Configuration > Configure Site**.
2. In the **Application Server Instances and Enabled Product Options** pane, select the service providers (for example, `WorkflowServer` and `UserServer32`) that you want to transfer back to `SOAServer`, and then drag and drop them. Repeat this process until all service providers have been moved.
3. For each instance, select the **Delete on save** check box.
4. Click **Save**.

All services will stop running, the service providers will be reconfigured, and the unused server instances and their services will be removed. The remaining services will then start again.

Multiple Server Installation

Multiple Server Installation on Multiple Machines

After installing the application server to host the Core services on one machine, you can set up additional machines to host extension servers in the form of a Workflow server, for the Workflow services, and one or more User servers, for product options that you may license and/or custom service providers that you develop.

Installing Workflow on extension servers involves two main steps:

1. Export security certificates
2. Install the extension server(s)

Prerequisites for Windows 2012 R2 Installations

To install the Application Server, Reporting, Extension servers, or ActiveX Task List on Windows[®] Server 2012 R2, the April 2014 update rollup for Windows[®] Server 2012 R2 (KB2919355) must already be applied.

**CAUTION:**

If this update is missing, the Visual C++ 2015 Redistributable packages installed by the Workflow installer will fail. These packages must be manually uninstalled, and then the necessary Windows updates must be applied before you attempt the Workflow installation again. If you do not uninstall the Visual C++ 2015 Redistributable packages before retrying the Workflow installation, the installation will appear to be successful but the WorkflowServer service will fail to start.

**Important:**

These recovery steps also apply if you attempt to install Workflow 2.5 SP2 on Windows[®] 7 Professional without SP1 or Windows[®] Server 2008 R2 without SP1.

Export server security certificates for an extension server

After installing the main application server in a multiple server environment, you must export the server security certificates and install them on your extension servers. During the extension server installation process, you must import these security certificates.

Procedure

1. On the main server computer, click **Start > All Programs > General Electric > Workflow > Configuration > Export Certificates**.
The **Export Workflow Application Server Certificates** page appears.
2. In the **Export Folder** field, enter the folder path or browse to the location where the certificates are to be exported.
3. In the **Password** field, enter a password to secure the exported certificates, which will be used when importing the certificates to the extension servers.
4. Click **Save**.
A file with the extension .zip is created, which contains the certificates required on the extension servers.
5. Click **Exit**.

Install Multiple Servers

This application provides multiple server capability, allowing you to install the application server to host the core services on one machine and a Workflow server to host the Workflow services on another machine. You can also choose to install one or more User servers to host other product options that you may license and/or custom service providers that you develop.

Before you begin

Before you can install multiple servers, you must install a single application server and client and export server security certificates for an extension server.



Important:

- If you are installing or upgrading an extension server (Workflow or User), you must have already installed or upgraded an application server of the same version to host the Core services on a separate machine.
- In a multiple server environment, eSOP is automatically installed on the Workflow server. When you upgrade Workflow or install it again, all sample resource files are overwritten.

Prerequisites

[Install a single application server and client \(on page 79\)](#)

[Export server security certificates for an extension server \(on page 91\)](#)

Procedure

1. From the application splash screen, click **Install Workflow Client and Application Server**.



Note:

If the splash screen does not appear, run `InstallFrontEnd.exe` on the root directory of the installation folder.

The setup wizard appears.

2. Click **Application Server**.

The **Application Server Type** page appears.

3. Click **Extension server**, select one of the following options, and then click **Next**.

To install a...	Click...
dedicated workflow server	Workflow Server
server to host custom service provider	User Server

The **License Agreement** page appears.

4. Review the license agreement, and then click **I Agree**.

The **Installation Folders and Architecture** page appears.

5. Proceed as applicable based on the architecture of the computer you are installing to:

If the architecture is...	Then do this...
32-bit	<p>a. Accept the default destination folder or browse for a new location, and then click Next.</p> <p>The Database Configuration page appears.</p>
64-bit	<p>a. Accept the default Destination Folder (32-bit), or browse for a new location.</p> <div data-bbox="938 653 1424 968" style="border: 1px solid #00a0c0; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note:</p> <p>This is the installation location for 32-bit components. Do not install 32-bit components to the 64-bit installation directory. If you do, the installation will fail.</p> </div> <p>b. Accept the default Destination Folder (64-bit), or browse for a new location.</p> <p>c. Select either 32-bit or 64-bit, depending on the architecture in which you want the server to operate.</p> <div data-bbox="938 1247 1424 1604" style="border: 1px solid #00a0c0; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Tip:</p> <p>The default is 64-bit, but you may also select 32-bit. If your environment changes after installation, you can use the Configure Server tool to change the architecture.</p> </div> <p>d. Click Next.</p> <p>The Database Configuration page appears.</p>

6. Specify your SQL Server settings, as follows:



Note:

These settings must be the same as for the core server.

- a. In the **Server** field, enter or select the name of the SQL Server to which you want to connect.
- b. In the **Database** field, enter the name of your SQL Server database, or click the drop-down arrow to search for all databases located on the specified server.
- c. From the **Authentication** list, select the type of authentication that you want to use, and then proceed based on your selection.
 - If you select **Windows Authentication**, proceed to step 7.
 - If you select **SQL Server Authentication**, proceed to step 6d.
- d. Enter the **User Name** and **Password** that are configured for SQL Server authentication, and then proceed to step 7.

7. Click **Next**.

The **Security Certificates** page appears.

8. Import the security certificates:



Important:

If you have not already exported the security certificates from the main server, you must do so now before proceeding. For more information, see **Export server security certificates for an extension server**.

- a. In the **Certificate File** field, enter the path or browse to the location where you exported the security certificates from the main server.
- b. Select the ZIP file that you created on the main server, and then click **OK**.
- c. In the **Password** field, enter the password that was set when you exported the security certificates.

9. Click **Next**.

The imported certificates are listed.

10. Click **Next**.

If the installation computer has an enabled firewall, the **Firewall** page appears.

11. If required, make note of the port numbers to open in the firewall, and then click **Next**.

**Important:**

If your system uses a firewall, you must follow the instructions on the Firewall page and open ports 8447, 8012, 8111, 8112, 8020, 8201, 8202, 8203, and 8204 to incoming TCP traffic. For information on opening a port in your firewall, see the online help for your Windows operating system.

The **Installation Confirmation** page appears.

**Tip:**

At any time during the configuration process, you can click Back to return to a previous page to change your settings.

12. Click **Install**.

The **Installation** page appears, displaying the status of each installation step.

**Note:**

In certain situations, a system restart is required after installation. If this is the case, a corresponding message is displayed among the listed status messages.

13. When the installation is complete, click **Exit**.

**Note:**

If the installation fails, or you want to change a configuration setting after the installation completes, you can open the appropriate standalone component to configure the required information. For more information, see **Post-Installation Configuration**.

Related information

[Post-installation Configuration \(on page 133\)](#)

One-click Deployment

One-click deployment allows you to deploy Workflow to users without running a client installation on their computers. It also provides the ability for a client to update itself when the server is upgraded.

 **Important:**

Support for one-click installation of Workflow will be discontinued in a future release of the software. The client installation should be used when one-click installation is discontinued.

Before you can use one-click deployment, you must complete the following prerequisites:

- Microsoft® .NET Framework 4.5 (Full Framework)

**Note:**

Custom display or form assemblies that target earlier versions of .NET will continue to function as before. However, saving changes made to such custom assemblies requires that they be upgraded, which is accomplished in different ways, based on the version of .NET:

- For an assembly that pre-dates .NET 4.0, you must upgrade the assembly when you open it. Failure to do so results in an error and the inability to save the edited assembly.
- For an assembly that targets .NET 4.0, the assembly is upgraded automatically when opened, and changes to such assemblies can be saved as before.

Service providers that target versions of the .NET framework before version 4.5 must be recompiled targeting .NET 4.5.

- The NetTCPPortSharing service must be enabled and started.
- Security certificates cannot be deployed remotely. Install your certificates by manually installing a copy of the certificates from the server. Certificates can be copied from the server to a disk or other portable device, and then installed, or by using your IT department's method of delivering and installing files.
- MIME types must be added to each computer in order for the `.manifest` and `.deploy` files to work correctly. For more information, see the following:
 - [http://msdn.microsoft.com/en-us/library/ms228998\(VS.85\).aspx](http://msdn.microsoft.com/en-us/library/ms228998(VS.85).aspx)
 - <http://msdn.microsoft.com/en-us/library/ms752346.aspx>

When an administrator installs Workflow on a server computer, the files required for the one-click deployment are installed when you perform a server installation.

On 32-bit computers, these files are located in a folder called `Deployment` in the following location:

```
<install dir>\Program Files\Proficy\Proficy Workflow\Program\Deployment
```

On 64-bit computers, these files are located in a folder called `Deployment` in the following location:

```
<install dir>\Program Files (x86)\Proficy\Proficy Workflow\Program\Deployment
```

An Administrator user must share this folder on the server computer, and then users can access it through a web page. The file used for one-click deployment is:

```
ProficyClient.application
```

You can create a web page, a button, or any means of access you want to use to make one-click deployment available to your users.

Each time a user invokes one-click deployment, it checks for any code updates, such as patches and service packs, that have been applied to the server (and subsequently to the one-click deployment files) and applies those changes to the one-click client. This allows your users to always be working with the same version of code as the server.

After installing Workflow on the server computer, you can copy the `Deployment` folder to another location for users to access. If you copy the folder to another location, you must re-copy the folder whenever code updates are applied in order for users to get the updates.

**Important:**

Do not move the `Deployment` folder from its install location. If this folder is moved, code updates cannot be applied to it.

Configure security certificates for a click-once client

For a click-once client, use this procedure to install certificates using the certificate authority generated by the Workflow server installation.

About this task

**Note:**

Support for one-click installation of Workflow will be discontinued in a future release of the software. The client installation should be used when one-click installation is discontinued.

Procedure

1. In the Workflow installation directory, go to the following folder to find the required certificate files:
`<installdir>\Program Files\Proficy\Proficy Workflow\Certificates\Export.`
2. Copy the following certificate files to your click-once computer:
 - ProficySelfSignedCA.pfx
 - ProficySTS.pfx
3. For a click-once client running on Windows, run `mmc.exe`.
4. From the **Console** window, go to the **File** menu, and then select **Add/Remove Snap-in**.
The **Add or Remove Snap-ins** dialog box appears. In Available Snap-ins, double-click Certificates.
5. In **Available Snap-ins**, double-click **Certificates**.
The **Certificates snap-in** dialog box appears.
6. Select **Computer account**, click **Next**, select **Local computer**, and then click **Finish**.
7. Click **OK**.
8. From **Certificates (Local Computer)**, expand Trusted Root Certification Authorities
`\Certificates`, and then add `ProficySelfSignedCA.pfx`.
9. From **Certificates (Local Computer)**, expand Personal\Certificates, and then add
`ProficySTS.pfx`.

Related reference

[SSL Security Configuration \(on page 1703\)](#)

[Security Certificate Options \(on page 59\)](#)

Related information

[Security Certificates \(on page 58\)](#)

[Modify security certificates \(on page 135\)](#)

[Install a single application server and client \(on page 79\)](#)

[Configure security certificates for a reverse proxy \(on page 116\)](#)

Display the host server name in the Proficy client for a one-click client

By default, the server name is displayed in the client, next to the server state icon. However, if you want to display a more meaningful name for your server, you can configure an alias so that the alias is displayed in the client rather than the actual server name.

Before you begin

You must install Workflow 2.5 SIM3 (or greater) before performing this task.

Procedure

1. On the click-once Workflow client computer, navigate to the click-once installation folder; for example, `C:\Users\\AppData\Local\Apps\2.0\ZJZ3X90C.HGM\9A05RKG3.TKC\prof..tion_15359f3836613cd4_0002.0005_14ea60ef7e3d4b0f\Data.`



Note:

The installation folder can be found in the updater log.

2. Open the `UserStartup.dita` file in a text editor.

The following code shows samples of multiple server names and aliases. You can copy this code into the `UserStartup.dita` file, before the closing `</ServerConnectionInfo>` element, and then change the `Name` and `Alias` elements to reflect your server information.

```
<ServerAliases>
  <ServerAlias>
    <Name>localhost</Name>
    <Alias>QA Server</Alias>
  </ServerAlias>
  <ServerAlias>
    <Name>Product-WIN81</Name>
    <Alias>Development Server</Alias>
  </ServerAlias>
  <ServerAlias>
    <Name>devwin7ex64dev</Name>
    <Alias>Product Server</Alias>
  </ServerAlias>
</ServerAliases>
```

When you launch the Workflow client, the alias that you configured for the server is displayed next to the state



Command-line (Silent) Installation

You can install the server, client, or hosts using command line parameters. You can also use this method to install multiple servers, ensuring that the same configuration data is used in all instances.

Command-line Parameters



Note:

Starting with Workflow 2.5, legacy certificates are no longer supported. For new server certificates, select the Generate new and unique certificates option.

Command Line Parameter	Description
<p>/s</p>	<p>Runs the installation setup (<code>SetupApplication-Server.exe</code> or <code>SetupWebServer.exe</code>) in silent mode; that is, the installation occurs without displaying the user interface.</p> <div data-bbox="834 758 1419 926" style="border: 1px solid #00a0c0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: The installation fails if the Proficy services are running. See /fs. </div> <p>You must use the /c parameter in conjunction with the /s parameter and provide a valid configuration file name.</p>
<p>/fs</p>	<p>Same as the /s parameter, but stops and restarts running services when performing an upgrade installation.</p>
<p>/t filepath (where <code>filepath</code> is the user-defined file name)</p>	<p>Runs the installation setup (<code>SetupApplication-Server.exe</code> or <code>SetupWebServer.exe</code>), including the user interface, and creates a configuration file that contains all of the installation configuration data.</p> <div data-bbox="834 1514 1419 1682" style="border: 1px solid #00a0c0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: For security reasons, passwords are not included in the configuration data. </div>
<p>PropertyName=value (where PropertyName is a configuration file property)</p>	<p>Individual configuration file properties, such as passwords, can be set using a command line by providing the property name and value separated by an equals sign (=). The property name cannot</p>

Command Line Parameter	Description
	<p>have spaces and values with spaces must be enclosed in double quotes. The following rules apply:</p> <ul style="list-style-type: none"> • parameter name matching is not case sensitive • double quotes around values are required to preserve white space (for example, PropertyA="a value with spaces") • command-line parameter values supersede property values supplied in a configuration file • double quotes in values are permitted, but must be doubled (for example, PropertyB="a ""quoted"" value") <p>Examples: Setting Password Properties</p> <p>Include one or more of the following on the command line to set the respective passwords to the value, secret.</p> <ul style="list-style-type: none"> • AdministratorPassword=secret • KspUserPassword=secret • CertificatePassword=secret • CertificateServicePassword=secret
<p>/c filepath (where <i>filepath</i> is the user-defined file name)</p>	<p>Runs the installation setup (<code>SetupApplication-Server.exe</code> or <code>SetupWebServer.exe</code>) and uses the configuration data found in the configuration file. When used with the /s parameter, the installation runs in silent mode.</p> <div style="border: 1px solid #00a0c0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note:</p> <p>You can use the configuration file that was created using the /t command line parameter or you can use the sample configuration file provided in the install directory.</p> </div>

Command Line Parameter	Description
	 If you use the configuration file created by using the /t parameter, you must add the passwords before using that file with the /c parameter.
/l filepath (where <code>filepath</code> is the logging directory)	Overrides the default location for the installation log file.
/p installation type option (where <code>installation type option</code> values are specific to the product being installed)	<p>Runs <code>SetupApplicationServer.exe</code>, including the user interface, without displaying the installation type selection screen, instead automatically selecting the installation type based on the value specified for the command line parameter. Valid parameter values based on product type are as follows:</p> <p>SetupApplicationServer.exe</p> <ul style="list-style-type: none"> • Server: Installs a application server in Single Server mode. • WorkflowServer: Installs a application server in Workflow Extension Server mode. • UserServer: Installs a application server in User Extension Server mode. • Client: Installs a Workflow client. <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note:</p> <ul style="list-style-type: none"> • These values are case-sensitive. • The /p parameter cannot be used in conjunction with silent mode (that </div>

Command Line Parameter	Description
	 is, the /s parameter) and/or a configuration file (that is, the /c parameter).

Uninstall Workflow

The following topic guides you through the process of removing Workflow software from your system.

About this task



Note:

If you generated security certificates or used pre-existing certificates from a previous version of this application, those certificates are not removed when you uninstall your application server or remote client.

Procedure

1. From **Control Panel**, go to your programs listing.
2. If applicable, select **ADAM Instance SOAAdam**, and then click **Uninstall** (depending on your operating system).
3. In the confirmation message box, click **Yes**.
4. Select *Workflow (<installation type>)*, and then click **Uninstall** (depending on your operating system).
5. In the confirmation message box, click **Yes**.

Server Clustering Implementation

Supported and Validated Cluster Configurations

Before you begin installing Workflow, it is important to determine your architecture so you can know where to install your servers.

There are three environments that have been validated and are supported.

- One cluster, two nodes
- Two clusters, two nodes each
- Three clusters, two nodes each

The only supported configuration of cluster groups is as follows.

- Application server (single server) cluster group
- Application server, Workflow Engine (multi-server) cluster group
- Application server, User Server (multi-server) cluster group
- Application server, Workflow Engine, and User Server (multi-server) clustered groups

Connecting the web server to the application server in a failover cluster environment is not currently supported.

Related reference

[Server Clustering and Failover Hardware Requirements \(on page 39\)](#)

[Server Clustering and Failover Software Requirements \(on page 32\)](#)

Related information

[One Cluster, Two Nodes \(on page 104\)](#)

[Two Clusters, Two Nodes Each \(on page 105\)](#)

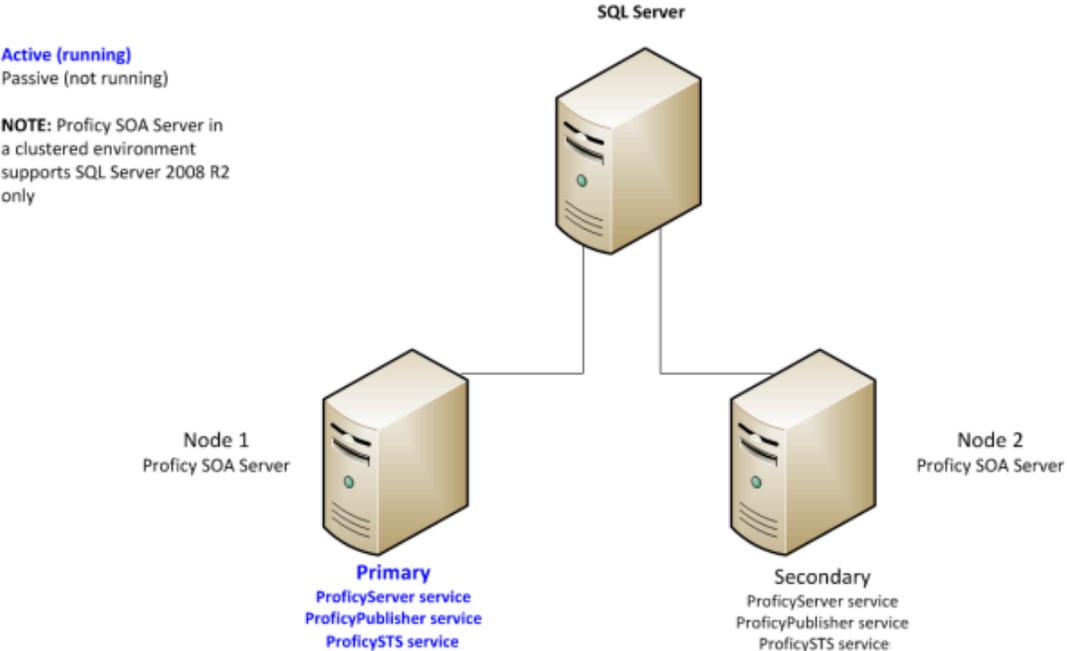
[Three Clusters, Two Nodes Each \(on page 107\)](#)

One Cluster, Two Nodes

One of the environments supported and validated for server clustering is a single cluster with two nodes.

The following figure shows an example of two application servers in one cluster. Each application server must run the Proficy Server, Proficy Publisher Service, and Proficy STS services.

Figure 1. Two Application servers in a single cluster

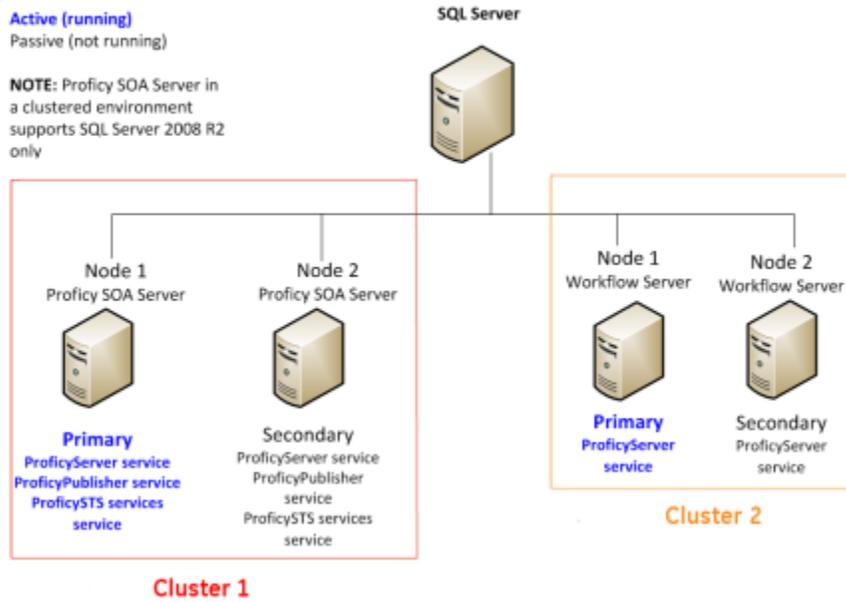


Two Clusters, Two Nodes Each

One of the environments supported and validated for server clustering is two clusters with two nodes each.

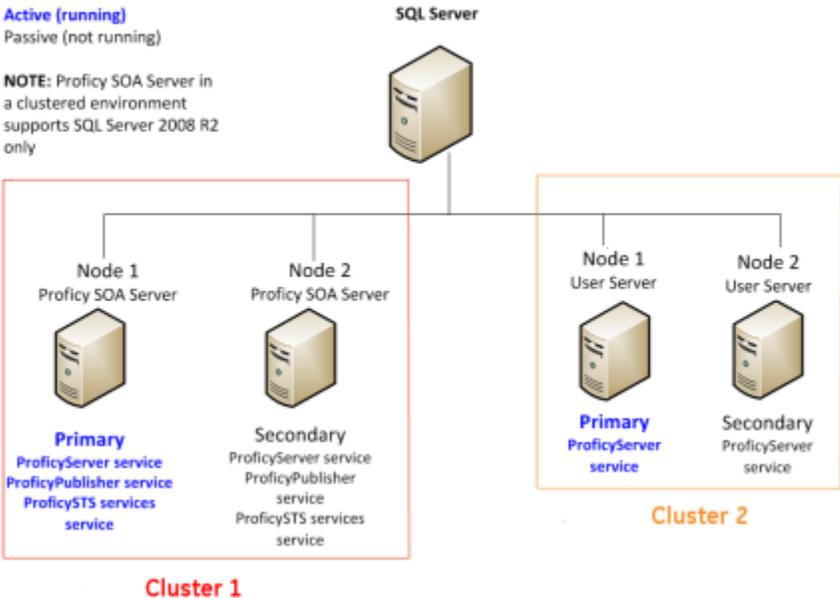
The following figures show examples of two application servers with two Workflow engines, in separate clusters, and two application servers with two User servers, in separate clusters, respectively.

Figure 2. Two Application servers and two Workflow engines, in separate clusters



Each application server must run the Proficy Server, Proficy Publisher Service, and Proficy STS services.
Each Workflow engine must run the Proficy Server service.

Figure 3. Two Application servers and two User servers, in separate clusters



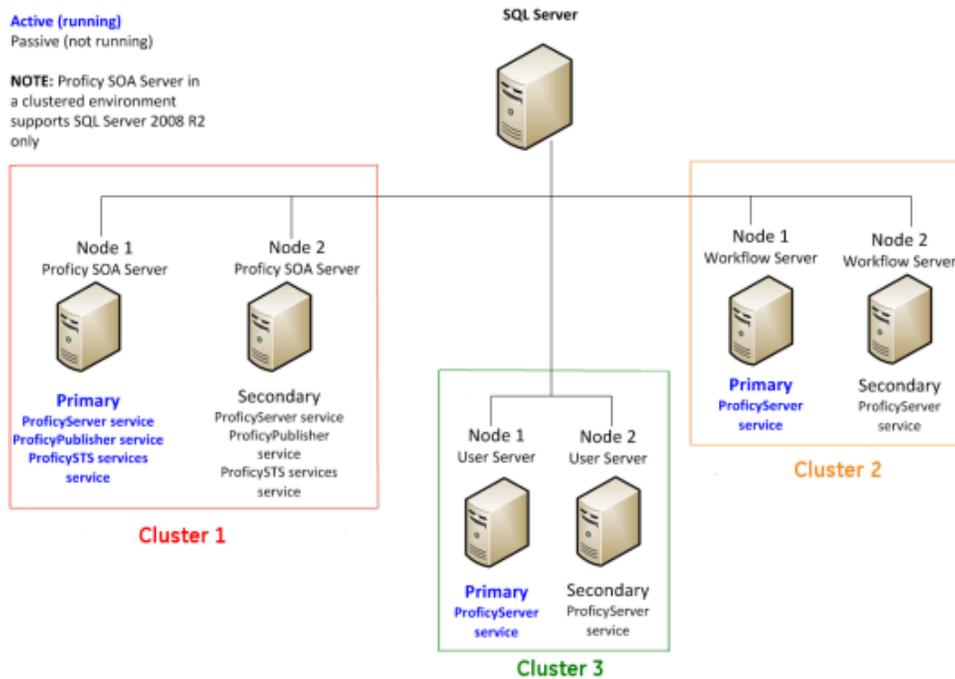
Each application server must run the Proficy Server, Proficy Publisher Service, and Proficy STS services. Each User server must run the Proficy Server service.

Three Clusters, Two Nodes Each

One of the environments supported and validated for server clustering is three clusters with two nodes each.

The following figure shows an example of two application servers, two Workflow engines, and two User servers, each in its own cluster. Each application server must run the Proficy Server, Proficy Publisher Service, and Proficy STS services. Each Workflow engine and each User server must run the Proficy Server service.

Figure 4. Two Application servers, two Workflow engines, and two User servers, in separate clusters



Microsoft Failover Cluster Manager for Windows Servers

The Microsoft Failover Cluster Manager is used to create the cluster and to add nodes to the cluster.

If failover clustering is not installed, use the **Server Manager** to install it.



Note:

Depending on your server requirements, visit the related Microsoft TechNet web sites for more information.

The Microsoft Failover Cluster Manager is used to create two new DNS entries: one to manage your server cluster, and one for the services you intend to cluster. For a application server cluster (even in a multi-server environment), you must add the Proficy Server, Proficy Publisher Service, Proficy STS, and Proficy Certificate services for clean installations, as well as upgrades.

If you are using clusters in a multi-server environment and clustering Workflow and/or User servers, only the Proficy Server service needs to be added for those servers.

Install Microsoft Failover Cluster Manager

The Microsoft Failover Cluster Manager must be installed in order to configure your failover cluster.

Procedure

1. On the server computer, click **Start > Administrative Tools > Server Manager**.
The **Server Manager** dialog box appears.
2. To configure the related server, select **Add roles and features**.
The **Add Roles and Features Wizard** appears.
3. Select **Features > Failover Clustering > Next**.
4. Click **Install**.
5. When the installation completes, click **Close**.

Configure the Microsoft Failover Cluster Manager

The Microsoft Failover Cluster Manager is used to create a new DNS entry to manage your server cluster.

Before you begin



Important:

You must have installed application server on both cluster nodes, pointing to the same database, before performing the following procedure.

Prerequisites

[Install Microsoft Failover Cluster Manager \(on page 109\)](#)

[Install a single application server and client \(on page 79\)](#)

About this task

This procedure is based on a Windows[®] Server 2012 R2 environment. If you are using Windows 2008 R2, ensure that you add the Proficy Publisher Service and Proficy STS services as children of Proficy Server service.

Procedure

1. On the server computer, click **Start > Administrative Tools > Failover Cluster Manager**.
The **Failover Cluster Manager** dialog box appears.
2. From the **Actions** menu, select **Create Cluster**.
The **Create Cluster Wizard** appears.
3. Follow the wizard prompts to create your cluster, as follows:

- a. Enter a management name for the cluster.
 - b. Add the nodes that will be part of the cluster; that is, the primary and failover servers.
 - c. In the **Failover Cluster Manager**, expand the cluster from the left hand side, and then click **Roles**.
 - d. Click **Configure Role**.
The **High Availability Wizard** appears.
 - e. Click **Generic Service**, and then click **Next**.
 - f. Click **Proficy Server**, and then click **Next**.
 - g. Enter a name for the server. This name is required when running various GE software tools.
 - h. Click **Next**, and then **Next** again to bypass the storage and registry settings.
 - i. Review the confirmation, click **Next**, and then click **Finish**.
4. To add support services, perform the following procedure.
- a. In the **Roles** window, in the **Status** column, wait until the status changes to *Running*.
 - b. Right-click the role that you created, click **Add Resource**, and then select **Generic Service**.
 - c. Click Proficy STS, and then click **Next**.
 - d. Review the confirmation, click **Next**, and then click **Finish**.
 - e. For Proficy Publisher Service, repeat steps b-d.
The **Status** indicates that the roles are *Partially Running*.
 - f. For Proficy Certificate, repeat steps b-d.
 - g. In the **Roles** window, select the name you entered in step 3g.
 - h. Click **Start Role**.
The **Status** indicates that the roles are *Running and Online*.
 - i. Click **Stop Role**.

What to do next

After setting up the failover cluster, the services will be running on both servers. You must either manually stop the services on the failover server, or move the service to the primary server. This enables the services on only the primary server.

**Note:**

To run any post-installation configuration tools, you must turn off the cluster to avoid a failover. For all tools, except **Configure Server**, you are required to run the tools on one node only of the cluster. For configuring the server, you must run the tool on all nodes.

Related information

[Post-installation Configuration \(on page 133\)](#)

[Configure a server instance \(on page 143\)](#)

Configure quorum options for a cluster

Although this application does not require a disk for clustering, it is recommended that you configure either a file share witness or a disk witness.

Procedure

1. On the server computer, click **Start > Administrative Tools > Failover Cluster Manager**.
The **Failover Cluster Manager** dialog box appears.
2. Select the cluster you want to configure.

**Tip:**

If the cluster you want to configure is not displayed, then in the console tree, right-click **Failover Cluster Manager**, click **Manage a Cluster**, and then select or specify the cluster you want to configure.

3. From the **Actions** menu, click **More Actions**, and then click **Configure Cluster Quorum Settings**.
4. Follow the instructions in the wizard to select the quorum configuration for your cluster.
If you choose a configuration that includes a disk witness or file share witness, follow the instructions for specifying the witness.
5. **Optional:** After the wizard completes and the **Summary** page appears, click **View Report** to view a report of the tasks that the wizard performed.

Configure the primary cluster server

After installing the application server on your primary server machine, you must configure the machine as the primary cluster server to begin the clustering process.

Prerequisites

[Install a single application server and client \(on page 79\)](#)

Procedure

1. On the primary server machine, On the primary server machine, click **Start > All Programs > General Electric > Workflow > Configuration > Configure Server**.
The **Configure Server** tool appears.
2. Select the Create an Application Server Instance option.
3. Select the **Make this machine a primary for a new failover cluster** check box.
4. In the **Cluster Name** field, enter the name of the cluster role (not the name of the cluster itself) you created in the Microsoft Failover Cluster Manager.
5. Click **Save**, and then click **Exit**.
6. Regenerate the security certificates for the cluster by running the Configure Proficy Service Certificates tool, and choose the **Generate new certificates unique to this installation** option.

What to do next



Important:

After configuring the primary cluster server, you must complete the following procedures to finish the clustering process.

1. Export the server security certificates in preparation for installing them on your failover and extension servers.
2. Configure the Failover Cluster Server.

Related information

[Modify security certificates \(on page 135\)](#)

[Export server security certificates for an extension server \(on page 91\)](#)

[Configure the failover cluster server \(on page 113\)](#)

Configure the failover cluster server

After configuring the primary application server cluster server and exporting the server security certificates, you can configure the failover cluster server.

Prerequisites

[Install a single application server and client \(on page 79\)](#)

[Configure the primary cluster server \(on page 112\)](#)

[Export server security certificates for an extension server \(on page 91\)](#)

Procedure

1. On the backup server machine, click **Start > All Programs > General Electric > Workflow > Configuration > Configure Server**.

The **Configure Server** tool appears.

2. Select the **Make this machine a failover backup of an existing Server Instance** option.
3. From the **Choose existing cluster** drop-down list, select the cluster that you want this machine to be the failover backup for.
4. Click **Save**, and then click **Exit**.

5. On the backup server machine, click **Start > All Programs > General Electric > Workflow > Configuration > Configure Certificates**.

The **Security Certificates** page appears.

6. Select the **Import certificates** option, and then import the certificates, as follows:
 - a. In the **Certificate File** field, click **Browse**, and then locate and select the .zip file containing the security certificates that you exported when you configured the primary server in the cluster.
 - b. In the **Password** field, enter the password that you set when you exported the security certificates from the primary server.
 - c. Click **Next**.
The page changes to indicate that the files contained in the exported certificate file have been loaded.
 - d. Click **Save**.

Configure a local client for a cluster node (32-bit)

If you want to run a cluster node on a 32-bit system, follow this procedure.

Before you begin**Tip:**

If your Workflow client starts but tries to update its product options and fails, then you must restart the Proficy Server service. Restarting this service ensures that the client product options are upgraded.

Prerequisites

[Install a single application server and client \(on page 79\)](#)

Procedure

1. To set up a local client, perform the following procedure.
 - a. Create a copy of the file, `UserStartup.dita`. To find this file, go to `<install location>\Program Files (x86)\Proficy\Proficy Workflow\Program\Data\UserStartup.dita`.
 - b. Name the new copy `UserStartupClient.dita`.
 - c. In the new file, change `ServerDNSName` to the name of the new cluster. For example, if it is called `soaCluster2012`: `<ServerDNSName>soaCluster2012</ServerDNSName>`.
2. In the file, `ProficyClient.exe.config`, make these changes. To find this file, go to `<install location>\Program Files (x86)\Proficy\Proficy Workflow\Program\ProficyClient.exe.config`.

**Note:**

It is recommended that you make a backup copy of this file before proceeding.

- a. From the line `<add key="stsUrl" value="http://<computer node name>:8112/ProficySTS">`, change **<computer node name>** to the name of the cluster, such as `soaCluster2012`.
- b. In the following two lines, change the old file name (`UserStartup.dita`) to the new file name (`UserStartupClient.dita`).
 - `<add key="url" value="./Data/UserStartupClient.dita">`
 - `<add key="UserStartupUrl" value="./Data/UserStartupClient.dita">`

Configure a local client for a cluster node (64-bit)

If you want to run a cluster node on a 64-bit system, follow this procedure.

Prerequisites

[Install a single application server and client \(on page 79\)](#)

Procedure

Point the file, `UserStartup.dita`, to the cluster instead of the node.

To find this file, go to `<install location>\Program Files (x86)\Proficy\Proficy Workflow\Program\Data\UserStartup.dita`.

Server Clustering and One-Click Deployment

If you have implemented a server cluster environment and you want to use one-click deployment in that environment, additional configuration is required to set up the one-click deployment.

There are two methods to choose from to configure one-click deployment in a server cluster environment.

- Modify the `userstartup.dita` file on each client to use the cluster address.
- Update the `userstartup.dita.deploy` file on the server, in the Proficy Workflow `\Program\Deployment\1_5_0_0\Program\Data` directory, and then regenerate the one-click deployment install.

Related information

[One-click Deployment \(on page 95\)](#)

Clustering Tips

Use the clustering tips to ensure your failover cluster runs smoothly.

Security Certificates

When setting up a clustered environment, you must regenerate and export Proficy security certificates as part of configuring the primary cluster server, and then import the certificates when configuring the failover cluster server. Symptoms of incorrect security certificate configuration include the inability to create or display Silverlight forms and the Web Task List. For more information on setting up security certificates for a clustered environment, see **Configure the Primary Server Cluster**, **Configure the Failover Cluster Server**, and **Modify Security Certificates**.

Proficy Server Service is Marked for Deletion

Occasionally, after a server configuration, the Proficy Server service is marked for deletion. You can restore the service by running the **Configure Server** tool again.

Clients on Clustered Proficy Servers

Clients that are installed with clustered Proficy servers will work only if the server is the primary server.

**Note:**

If you have a cluster with multiple nodes, the cluster must be pointing to the client name, and the server that each node is running on must state the role name for the cluster. For more information, see **Configure a local client for a cluster node (32-bit)** and **Configure a local client for a cluster node (64-bit)**.

Running a Clustered Multi-server Environment

When running with both a clustered application server and clustered Workflow/User servers, you must restart the primary server and then, after a few minutes, restart the primary Workflow/User server. This can be performed using the Microsoft Failover Cluster Manager by taking the servers offline, and then putting them back online.

Deploy the Web Task List

Web Task List Deployment

The Web Task List is automatically installed with the application server installation of Workflow. Users can access the Web Task List using a web browser.

A reverse proxy is highly recommended when deploying the Web Task List, as it provides increased security by limiting and controlling the exposure of internal servers and services. You can configure either an Internet Information Services (IIS) reverse proxy or an Apache reverse proxy for use with the Web Task List. In either case, you must also configure the necessary certificates before connecting to the Web Task List.

Configure security certificates for a reverse proxy

For a secure reverse proxy, use this procedure to install certificates to your web server using the certificate authority generated by the application server installation.

Procedure

1. In the Workflow installation directory, go to the following folder to find the required certificate files:
`<installdir>\Program Files\Proficy\Proficy Workflow\Certificates\Export.`
2. Copy the following certificate files to your web server:

- ProficySelfSignedCAPublicKey.cer
 - ProficySTSPublicKey.cer
3. For a web server running on Windows, run `mmc.exe`.
 4. From the **Console** window, go to the **File** menu, and then select **Add/Remove Snap-in**.
The **Add or Remove Snap-ins** dialog box appears.
 5. In **Available Snap-ins**, double-click **Certificates**.
The **Certificates snap-in** dialog box appears.
 6. Select **Computer account**, click **Next**, select **Local computer**, and then click **Finish**.
 7. Click **OK**.
 8. From **Certificates (Local Computer)**, expand **Trusted Root Certification Authorities** \ **Certificates**, and then add `ProficySelfSignedCAPublicKey.cer`.
 9. From **Certificates (Local Computer)**, expand **Personal\Certificates**, and then add `ProficySTSPublicKey.cer`.

Related reference

[SSL Security Configuration \(on page 1703\)](#)

[Security Certificate Options \(on page 59\)](#)

Related information

[Security Certificates \(on page 58\)](#)

[Modify security certificates \(on page 135\)](#)

[Install a single application server and client \(on page 79\)](#)

Configure an IIS reverse proxy for the Web Task List

The information provided in this section is specific to configuring Internet Information Services (IIS) to act as a reverse proxy using the Application Request Routing (ARR) and URL Rewrite modules.

Before you begin**Tip:**

Before installing IIS Application Request Routing (ARR), it is recommended that you enable HTTP Logging and Tracing features for IIS first. Tracing makes it possible to diagnose problems with reverse proxy rewrite rules.

**Note:**

Before you make any changes to Internet Information Services (IIS), you may want to export your existing configuration to review it. For more information on an existing Microsoft utility, refer to <https://www.iis.net/learn/get-started/getting-started-with-iis/getting-started-with-appcmdexe>

1. From the **Control Panel**, go to **Windows Features** to enable the following IIS settings:

- HTTP Logging
- Tracing

**Tip:**

To make it easier to diagnose problems, tracing must be enabled before installing ARR.

- The ARR and URL Rewrite modules only install their trace providers if those IIS features are already enabled. You can install them later by enabling the IIS HTTP Logging and Tracing features and reinstalling or repairing ARR.
 - Tracing is disabled by default.
2. Install the IIS ARR 2.5 (or later) module from <http://www.iis.net/download/ApplicationRequestRouting>.
3. In the **Install** window, click **Options**.
4. In the **Change Options** window, when asked which web server you want to use, click the **IIS** check box.
5. Click **OK**, and then click **Install**.

**Note:**

The URL Rewrite 2.0 module installs automatically.

6. Start the Internet Information Service (IIS) Manager application.
7. From the **Start** menu, click **Administrative Tools > Internet Information Services (IIS) Manager**.
8. Enable the server to act as a proxy.

**Note:**

This is a server-wide setting.

- a. From the **Connections** panel, select the server you want to configure.
- b. From the **IIS** section in the central panel, double-click **Application Request Routing Cache**.

- c. From the **Actions** panel, in the **Proxy** section, click **Server Proxy Settings. Application Request Routing** opens in the central panel.
- d. Select the **Enable proxy** check box, and then in the **Actions** panel, click **Apply**.

The URL Rewrite rules apply to a single web site. These instructions and rules assume that your server already is configured with a web site, such as Default Web Site, with an HTTPS binding on port 443 that uses a certificate that is trusted by the client machines.

Procedure

1. In the Internet Information Services (IIS) Manager, in the **Connections** panel, under the server you are currently configuring, expand the **Sites** node, and then select the web site that you want to configure.
2. From the **Actions** panel, click **Explore** to open Windows Explorer to the physical directory on disk for your web site.
 - The default location for **Default Web Site** is `C:\inetpub\wwwroot`.
 - In this location, you see a file named `web.config`.
3. Open another Windows Explorer window and navigate to your Workflow installation directory.
 - On 32-bit systems, the default location is `C:\Program Files\Proficy\Proficy Workflow`.
 - On 64-bit systems, the default location is `C:\Program Files (x86)\Proficy\Proficy Workflow`.
4. In the Workflow installation directory, go to the folder **Reverse Proxy Config (IIS)** for the reverse proxy configuration files.
 - a. From the **Reverse Proxy Config (IIS)** directory, copy the following files to your web site's directory:
 - `rewriteMaps.config`
 - `rules.config`
 - `outboundRules.config`
 - b. For a sample of the `web.config` showing how to configure the **<rewrite>** element under **<system.webServer>**, go to the **Reverse Proxy Config (IIS)** directory.
If your web site's `web.config` has:
5. Update the `rewrite.Maps.config` file with the actual server names, as follows:

- a. In the Internet Information Services (IIS) Manager, in the **Connections** panel, under the server you are currently configuring, expand the **Sites** node, and then select the web site whose `rewriteMaps.config` file you want to update.
- b. Double-click the **URL Rewrite** icon.
- c. From the **Actions** panel, click **View Rewrite Maps**. **URL Rewrite Maps** opens in the central panel.
- d. Double-click **ComputerNames**. **Edit Rewrite Map** opens in the central panel.
- e. From the **Original Value** column, select a server, and then from the **Actions** panel, click **Edit Mapping Entry**. The **Edit Mapping Entry** dialog box appears.
- f. In the **New Value** field, enter the actual server name, as follows:
 - For the `WebServerHostName`, enter the fully qualified name of the computer where your Web Task List is installed.
 - For all other server values, enter the fully qualified name of the computer where your servers are installed.
- g. Repeat steps 5e and 5f for each server listed in the **Edit Rewrite Map** panel.

Related reference

[SSL Security Configuration \(on page 1703\)](#)

[Security Certificate Options \(on page 59\)](#)

Related information

[Install a single application server and client \(on page 79\)](#)

[Security Certificates \(on page 58\)](#)

[Configure security certificates for a reverse proxy \(on page 116\)](#)

[Configure an Apache reverse proxy for the Web Task List \(on page 120\)](#)

Configure an Apache reverse proxy for the Web Task List

The information provided in this section is specific to configuring Apache to act as a reverse proxy.

Before you begin

To perform this configuration, you must already be familiar with Apache.

About this task**Note:**

This procedure assumes that the application server and Workflow are co-resident on the same host.

Procedure

1. Enable these modules in `httpd.conf`:
 - `mod_proxy`
 - `mod_proxy_http`
 - `mod_substitute`
2. Add the rules from the **Inbound Rules** section to `httpd.conf`.
3. Add the rules from the **Outbound Rules** section to `httpd.conf`.

Related reference

[Inbound and Outbound Rules for Apache \(on page 121\)](#)

Inbound and Outbound Rules for Apache

Follow these rules when configuring an Apache reverse proxy for the Web Task List.

Inbound Rules

SSLProxyEngine On:		
ProxyPass	/Proficy/	http://YourSOAWorkflow.compa- ny.com:8020/Proficy/
ProxyPassReverse	/Proficy/	http://YourSOAWorkflow.compa- ny.com:8020/Proficy/
ProxyPass	/proficy/	http://YourSOAWorkflow.compa- ny.com:8020/Proficy/
ProxyPassReverse	/proficy/	http://YourSOAWorkflow.compa- ny.com:8020/Proficy/
ProxyPass	/ProficySTS/	https://YourSOAWorkflow.com- pany.com:8012/ProficySTS/
ProxyPassReverse	/ProficySTS/	https://YourSOAWorkflow.com- pany.com:8012/ProficySTS/

SSLProxyEngine On:		
ProxyPass	/SOAServer/SOAProject/	https://YourSOAWorkflow.com-company.com:8203/SOAServer/SOAProject/
ProxyPassReverse	/SOAServer/SOAProject/	https://YourSOAWorkflow.com-company.com:8203/SOAServer/SOAProject/
ProxyPass	/SOAServer/System/	http://YourSOAWorkflow.com-company.com:8020/SOAServer/system/
ProxyPassReverse	/SOAServer/System/	http://YourSOAWorkflow.com-company.com:8020/SOAServer/system/
ProxyPass	/WorkflowServer/	https://YourSOAWorkflow.com-company.com:8203/WorkflowServer/
ProxyPass	/WorkflowServer/	https://YourSOAWorkflow.com-company.com:8203/WorkflowServer/

Outbound Rules

AddOutputFilterByType SUBSTITUTE text/xml	
SUBSTITUTE	"s <SOAServerName>.*</SOAServerName> <SOAServerName> YourApacheServer.com </SOAServerName> i"
SUBSTITUTE	"s <StsServiceName>.*</StsServiceName> <StsServiceName> YourApacheServer.com </StsServiceName> i"
SUBSTITUTE	"s <StsSecurePort>.*</StsSecurePort> <StsSecurePort>443</StsSecurePort> i"
SUBSTITUTE	"s <HttpsSamlPort>.*</HttpsSamlPort> <HttpsSamlPort>443</HttpsSamlPort> i"
SUBSTITUTE	"s <WorkflowServerName>.*</WorkflowServerName> <WorkflowServerName> YourApacheServer.com </WorkflowServerName> i"

AddOutputFilterByType SUBSTITUTE text/xml	
SUBSTITUTE	"s <WorkflowSamlPort>.*</WorkflowSamlPort> <WorkflowSamlPort>443</WorkflowSamlPort> i"
SUBSTITUTE	"s <HttpPort>.*</HttpPort> <HttpPort>443</HttpPort> i"
SUBSTITUTE	"s (.*<a:anyURI>.*:.*?/(.*)</a:anyURI>(.*))\$1<a:anyURI>https://YourApacheServer.company.com:443/\$2</a:anyURI>\$3 i"

Related information

[Configure an Apache reverse proxy for the Web Task List \(on page 120\)](#)

Connect to the Web Task List

The information in this section guides you through the process of connecting to the Web Task List.

Before you begin

You can configure and use a dedicated reverse proxy for increased security when using the Web Task List over the Internet.



Tip:

To improve performance, ensure that caching is enabled in Internet Explorer.

Prerequisites

[Install a single application server and client \(on page 79\)](#)

[Configure an IIS reverse proxy for the Web Task List \(on page 117\)](#)

Procedure

When the Full Server installation and reverse proxy configuration have been completed, the Web Task List can be accessed using the following URLs.

Web Server	URL
IIS Reverse Proxy	https://myiisserver.example.com/Proficy/Workflow/WebTaskList.html

Web Server	URL
	<div data-bbox="834 275 1425 573" style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px;">  Note: Assumes reverse proxy IIS web server is externally accessible via host name <code>myi-isserver.example.com</code> with an SSL certificate specifying that host name listening on standard port 443 (default). </div>
<p data-bbox="363 600 634 632" style="text-align: center;">Apache Reverse Proxy</p>	<p data-bbox="818 600 1341 678">https://myapacheserver.example.com/Proficiency/Workflow/WebTaskList.html</p> <div data-bbox="834 716 1425 1014" style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px;">  Note: Assumes reverse proxy Apache web server is externally accessible via host name <code>myapacheserver.example.com</code> with an SSL certificate specifying that host name listening on standard port 443 (default). </div>
<p data-bbox="464 1052 537 1083" style="text-align: center;">Direct</p>	<p data-bbox="818 1052 1398 1129">http://mysoaserver:8020/Proficiency/Workflow/WebTaskList.html</p> <p data-bbox="818 1167 1370 1287">On the Full Server machine, a link to this URL is available from: Start menu > Proficiency > Proficiency Workflow > Task Lists > Web Task List.</p> <div data-bbox="834 1325 1425 1854" style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px;">  Note: The direct link is for intranet and testing only. We recommend using reverse proxy to use the Web Task List over the Internet to limit exposure to internal machines and services as part of a secure architecture. Assumes Workflow server is internally accessible via host name <code>mysoaserver</code> with an SSL certificate specifying that host name listening on standard port 8203 (default). A certificate is configured like this </div>

Web Server	URL
	 by default during a standard Full Server installation. The machine must also install the Proficy STS security certificate.

What to do next

Certificates

The machine running the Web Task List must trust the SSL certificates used by the server. If the SSL certificate is signed by a trusted root certificate authority (CA), then the client (that is, the Web Task List) machine must trust the signing CA by adding it to the Trusted Root Certificate Authority certificate store for the machine.



Note:

SSL certificates purchased from a reputable certificate authority will already be trusted.

For a direct connection, the Web Task List machine must also install the Proficy STS security certificate into the Personal certificate store for the machine.

For a reverse proxy configuration, the web server must trust the signing CA by adding it to the Trusted Root Certificate Authority certificate store for the machine. The Web Task List machine only needs to trust the SSL certificate for the web server and does not need to trust the SSL certificate used by the Workflow server of the Proficy STS security certificate.

Related reference

[SSL Security Configuration \(on page 1703\)](#)

[Security Certificate Options \(on page 59\)](#)

Related information

[Security Certificates \(on page 58\)](#)

[Configure security certificates for a reverse proxy \(on page 116\)](#)

[Configure an Apache reverse proxy for the Web Task List \(on page 120\)](#)

Log On Overview

Log On Overview

After installing Workflow, the primary administrator user configured during installation can manually log into the program and configure system security. As part of this process, additional users are specified and log in mode and authentication method are established.

Available log-in modes are manual and automatic; available authentication methods are Workflow users, Windows users, and SSO (GE Single Sign on).

Related information

[Security Concepts \(on page 334\)](#)

[Login and Logout Capabilities \(on page 384\)](#)

[Manual Login and Logout \(on page 395\)](#)

[Automatic Login and Logout \(on page 385\)](#)

[Windows Domain-Based Security \(on page 357\)](#)

[Multiple Sessions of Workflow \(on page 236\)](#)

Log on to Workflow client

You can log on to the Workflow client in one of two ways: automatic and manual. This procedure outlines the steps to manually log into the system.

Before you begin

To log in to Workflow, the system must be in a state of either *Complete* or *Partial* health. For more information, see **System Health and Program Use**.

Prerequisites

[Windows Domain-Based Security \(on page 357\)](#)

Procedure

1. Use one of the following methods to log on.

If...	Then do the following...
<p>An instance of Workflow is <u>not</u> running on the workstation.</p>	<p>a. Launch the Workflow client using either the Start menu or desktop icon.</p> <p>b. On the main Workflow screen, click Log in to Workflow.</p> <div data-bbox="938 590 1421 814" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note: You can also use this path if an instance of Workflow is already running on the workstation.</p> </div>
<p>An instance of Workflow is running on the workstation,</p>	<p>a. On the Workflow toolbar, click New Session.</p>

The **Workflow Login** dialog box appears.

2. In the **Authentication Type** section, select the type appropriate to your user profile.

Select...	If...
<p>Workflow Authentication</p>	<p>Your user profile is part of the Workflow application.</p>
<p>Windows Authentication</p>	<p>Your user profile is part of the Windows domain.</p> <div data-bbox="862 1371 1421 1814" style="border: 1px solid #FFD700; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Important: To log in using Windows Authentication, your server and client(s) must be on a domain. The domain field is populated using the domain that the computer joins. If the user logging in is from a different domain, this field is populated using the domain that this user belongs to.</p> </div>
<p>SSO Authentication</p>	<p>Your user profile is part of SSO security.</p>

Select...	If...
	 Note: SSO Authentication is not available if it has not been enabled.

3. In the **User Name** field, enter your user name.



Note:

If you are logging in using SSO Authentication, this field is labeled **User ID**.

4. In the **Password** field, enter your password.

5. If applicable, in the **Domain** field, enter the domain name.



Note:

To log in using Windows Authentication, your server and client(s) must be on a domain. The domain field is populated using the domain that the computer joins. If the user logging in is from a different domain, this field is populated using the domain that this user belongs to.

6. Click **OK**.

Related information

[Login and Logout Capabilities \(on page 384\)](#)

[Multiple Sessions of Workflow \(on page 236\)](#)

[System Health and Program Use \(on page 50\)](#)

Automatically log into Workflow

The default method of logging in to Workflow, and the only method available immediately after a new installation, is manually. However, the system can be configured to allow automatic login, in which case the steps for logging in are those outlined in this procedure.

Before you begin

To log in to Workflow, the system must be in a state of either *Complete* or *Partial* health. For more information, see **System Health and Program Use**.

Prerequisites

[Configure automatic login \(on page 390\)](#)

Procedure

1. Launch the Workflow client using either the **Start** menu or desktop icon.
If automatic login is enabled and configured, the user credentials are authenticated and the Workflow client appears.
2. If applicable, launch additional sessions of Workflow.
3. If you log out of any session for any reason, the **Login** dialog box appears. Click **Auto Login** to log in again with the appropriate credentials.

Related information

[Automatic Login Authentication \(on page 389\)](#)

[Windows Domain-Based Security \(on page 357\)](#)

[System Health and Program Use \(on page 50\)](#)

GE Single Sign On (SSO)

You can configure Workflow to use GE Single Sign On (SSO) authentication when your users log on to the Workflow client.



Note:

GE Single Sign On applies only to GE businesses.

Within GE, all employees are assigned an SSO ID that grants them access to internal GE web pages and resources. When Workflow is installed within a GE business, you have the option to enable SSO authentication, so that users enter their SSO ID and password to access the system.



Note:

To log in to Workflow, the system must be in a state of either *Complete* or *Partial* health. For more information, see **System Health and Program Use**.

Configuration Settings

GE SSO authentication can be configured during the installation process, or at any time after installation using the **Configure Security** tool.

When setting up SSO authentication, you must select either the **Production Identity Provider** or **Non-Production Identity Provider** option. During a new installation or an upgrade, when you select the **Use SSO** (Single Sign On) check box, the **Production Identity Provider** option is selected by default.

The Production Identity Provider option provides greater security for your application. The Non-Production Identity Provider can be used for non-production environments, such as test environments.

User Account Setup

After you configure your system for SSO authentication, you can manually set up each user's account to link to their SSO account. The account login name must be the user's SSO ID.

You can also use the SSO Authentication Failed event to schedule a workflow that will create a Workflow user account. That is, if a user with a valid SSO ID but no Workflow user account tries to log in to Workflow, the SSO Authentication Failed event is triggered and causes the Workflow user account to be created. The SSO Authentication Failed event contains the user's SSO ID, full name, and e-mail address.



Important:

Make sure that each user belongs to the SSO Users group before creating their SSO user account. The SSO Users group is a default security group located within the Personnel model.

Related information

[Workflow User Password Security \(on page 396\)](#)

[Install a single application server and client \(on page 79\)](#)

[Configure GE Single Sign on \(SSO\) \(on page 397\)](#)

[System Health and Program Use \(on page 50\)](#)

Authentication

The **Workflow Login** dialog box is used to log in to Workflow.

Authentication Types

The following information applies to all regular logins except the Mobile-sized Task List:

The computer you use to log in, stores the authentication type in the `proficyclient.exe.config` file. When you start a new client on that computer, the authentication type that you had selected during your previous login session is automatically selected. However, if another user had successfully logged in to that computer with a different authentication type between now and the previous time you had logged in, then that authentication type will be selected.

If you attempt to log in using a different authentication type but the authentication fails, the authentication type is not updated in the `proficyclient.exe.config` file.

Property	Value	Description
Workflow Authentication	System-defined	Your user profile is part of the Workflow application.
Windows Authentication	System-defined	<p>Your user profile is part of the Windows domain.</p> <div data-bbox="1024 611 1419 1236" style="border: 1px solid orange; border-radius: 10px; padding: 10px; background-color: #fff9c4;"> <p> Important: To log in using Windows Authentication, your server and client(s) must be on a domain. The domain field is populated using the domain that the computer joins. If the user logging in is from a different domain, this field is populated using the domain that this user belongs to.</p> </div> <p>You must have added your user accounts to one or more active directory universal or global groups that are mapped to Workflow groups.</p> <p>The computer stores the domain name you enter in the <code>proficyclient.exe.config</code> file. When you start a new client, that domain name appears in the Domain field. However, if another user successfully logged in under a different domain since your last</p>

Property	Value	Description
		<p>log on, that domain name will be displayed.</p> <p>If you attempt to log in under a different domain but the authentication fails, the domain name is not updated in the <code>proficiency-client.exe.config</code> file.</p>
SSO Authentication	System-defined	<p>Your user profile is part of SSO security.</p> <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: SSO Authentication is not available if it has not been enabled. </div>

Login Properties

Property	Value	Description
User Name	User-defined	Specifies your unique user name
Password	User-defined	<p>Specifies your unique password.</p> <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: Your password must conform to the defined password complexity rules. </div>
Domain	User-defined	<p>Specifies the Windows domain that your user profile is part of. This field is available when the Windows Authentication option is selected.</p>
Change Password	N/A	Click this link to open the Proficiency Workflow Change Password di-

Property	Value	Description
		<p>alog box where you can create a new password.</p> <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: This link is available only if this feature is enabled. </div>
Auto Login	System-defined	<p>Click this button to automatically log in the Workflow.</p> <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: This button is available only when the Windows Authentication option is selected <u>and</u> automatic login is configured. </div>

Post-Installation Configuration

Post-installation Configuration

This section provides information about configuring your system after installing the application.

If the installation fails, or you want to change a configuration setting after installation completes, you can open each of the installation wizard configuration pages as a standalone tool and make the required changes.



CAUTION:

Changes to your installation configuration should be performed only by advanced users. If you make changes on your own, unpredictable results and behavior may occur. Contact GE for assistance.

Running with a Standard Windows User Account

Users who are not Workflow administrators must be given permission to access certain folders in order to view log files and to be able to load forms.

- To provide the ability to load forms, ensure your users have permission to access C:\Program Data\Proficy\Logs.
- To provide access to log files, ensure your users have permission to access C:\Users\

When a form is opened, it is retrieved from the database and copied to the client computer to be loaded into Workfkow. If the **My Documents** folder is in a network storage location, that location is locked by user permissions, preventing the form from being created.

Configure web proxy settings for Local System account

Before defining a web service, you must configure web proxy settings for your Local System account based on your network requirements.

Before you begin

You may not need to change your network settings.

Procedure

1. Configure Internet web proxy setting using BITSAdmin in one of the following ways.



Note:

For more information on BITSAdmin Tool, go to: <http://msdn.microsoft.com/en-us/library/aa362813%28VS.85%29.aspx>

To set...	Then do this...
<p align="center">accounts to use a static proxy server with exclusions</p>	<p>a. Execute: <code>bitsadmin /util /setieproxy localsystem MANUAL_PROXY proxysrv:8080 " ;*.contoso.com"</code></p> <p>b. Replace proxysrv, 8080, and contoso.com with your organization's proxy server addresses address, port, and exclusions.</p>

To set...	Then do this...
account to use proxy.pac file	a. Execute: <code>bitsadmin /util /setieproxy localsystem AUTOSCRIPPT http://contoso.com/proxy.pac</code> b. Replace <code>proxysrv</code> with your organization's pac file addresses.

2. From an administrative command line window, run the commands for the account that you want to change.

Related information

[Add a web services definition \(on page 1577\)](#)

[Web Services Service Provider \(WSSP\) \(on page 1576\)](#)

[Install a single application server and client \(on page 79\)](#)

Modify security certificates

The **Configure Certificates** tool is used to change or update the security certificates you configured during the server installation.

Before you begin

Before you modify security certificates:

- Copy security certificates and transfer the copies to your extension servers.
- Update the security certificates to match the server.

About this task

Related Server Notes

- For an extension server installation (that is, a multiple server or server failover cluster), you must copy the ProficyPlatform and Proficy STS security certificates from the main server onto a disk or other portable device, and then transfer those certificate copies to your extension servers. The SSL/TLS Server Certificate must be unique to each server.
- If you change the certificate option on your servers, all remote clients connected to those servers must have their security certificates updated to match the server. Use the **Configure Client** tool to download the new versions of the certificates.

Upgrading

Beginning with Workflow 2.5, legacy certificates are no longer supported.

Procedure

1. On the server machine, On the server machine, click **Start > All Programs > General Electric > Workflow > Configuration > Configure Certificates**.
The **Security Certificates** page appears.
2. Select the appropriate certificate option. For full descriptions of the certificate options, see **Security Certificate Options**.
3. Click **Save**, and then click **Exit**.

Related reference

[Security Certificate Options \(on page 59\)](#)

Related information

[Install a single application server and client \(on page 79\)](#)

[Post-installation Configuration \(on page 133\)](#)

[Modify the server configuration for a remote client \(on page 140\)](#)

[Security Certificates \(on page 58\)](#)

[Use self-signed certificates for the web server \(on page 66\)](#)

Modify site configuration

Using the **Configure Site** tool, you can modify site configuration by moving product options between servers, deleting servers that are not hosting product options, and, beginning with Workflow 2.1, designating whether servers or the product options that run on them are essential and/or disabling product options.

Procedure

1. On the server computer, click **Start > All Programs > General Electric > Workflow > Configuration > Configure Site**.
The **Configure Site** tool appears.
2. Modify the server configuration, as required.

If you want to...	Then do this:
Designate a server or a product option as essential	Select the Essential check box beside the server or product option listing.

If you want to...	Then do this:
	<p> Note:</p> <p>Essential servers must remain running in order for users to access the program. If you designate a server as essential, all associated product options are automatically designated as essential. If you designate any product option as essential, the server that is hosting that option is automatically designated as essential, as indicated by the color filling in the Essential check box, but the essential status of other product options hosted by the server remains unchanged.</p>
<p>Designate a server or a product option as nonessential</p>	<p>Clear the Essential check box beside the server or product option listing.</p> <p> Note:</p> <p>If one or more product options assigned to a server is designated as essential, you cannot designate the server as nonessential. Nonessential servers must remain running in order for the program to be fully operational; however, if a nonessential server stops running, users can still access the program.</p>
<p>Move a product option between servers</p>	<p>In the Application Server Instances and Enabled Product Options pane, click the product option listing, and then do one of the following:</p>

If you want to...	Then do this:
	<ul style="list-style-type: none"> ◦ Click the Move Up or Move Down button until the listing is located below the server that you want to host the option. ◦ Drag and drop it below the server that you want to host the option.
Disable a product option	<p>In the Application Server Instances and Enabled Product Options pane, click the product option listing, and then do one of the following:</p> <ul style="list-style-type: none"> ◦ Click Disable. ◦ Drag and drop the listing to the Disabled Product Options pane. <div data-bbox="862 800 1424 1024" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note: Disabled product options remain installed but are deactivated. Essential product options cannot be disabled.</p> </div>
Enable a disabled product option	<p>In the Application Server Instances and Enabled Product Options pane, select the server that you want to host the product option, and then in the Disabled Product Options pane, click the listing for the product option and do one of the following:</p> <ul style="list-style-type: none"> ◦ Click Enable. ◦ Drag and drop the listing to the Application Server Instances and Enabled Product Options pane.
Remove a server	<p>Select Delete on save beside the server listing.</p> <div data-bbox="862 1583 1424 1732" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note: Only servers that are not hosting product options can be removed. To subse-</p> </div>

If you want to...	Then do this:
	 frequently add a server, use the Configure Server tool.

3. Click **Save**, and then click **Exit**.

What to do next

The designation of servers and production options as either essential or nonessential and the state and accessibility of servers has implications for the health of the site and, therefore, the ability to log in and use the Workflow program. For more information, see **System Health and Program Use**.

Related information

[Configure a server instance \(on page 143\)](#)

[System Health and Program Use \(on page 50\)](#)

Monitor system configuration and status

After installing this application, you can monitor the configuration of the system, including which servers and product options are installed and whether those components are designated as essential. In addition, you can ascertain the operational status of the configured servers.

Procedure

1. In the navigator, click **Proficy System > Proficy System**.
2. In the **Displays** panel, click **System Status**.

The **System Status** display appears in the workspace, listing the servers configured for the site. For each server, the display reflects whether the server is designated as essential; the essential and nonessential product options (or services) hosted on the server, if applicable; and whether the server is *Stable* or *Unreachable*.



Note:

The **System Status** display is read-only. However, you can use the **Configure Site** tool to modify site configuration, including designating whether servers or the product options that run on them are essential, moving product options between servers, disabling product options, and/or deleting servers that are not hosting product options.

Related information

[System Health and Program Use \(on page 50\)](#)

[Multiple Servers \(on page 68\)](#)

[Modify site configuration \(on page 136\)](#)

Modify the server configuration for a remote client

Use the **Server Configuration** tool to synchronize any changes you may have made to the application server with remote clients.

Before you begin



Important:

The **Server Configuration** tool is available only for client-only installations.

Procedure

1. On the client computer, click **Start > All Programs > General Electric > Workflow > Configuration > Configure Client**.
The **Server Configuration** tool appears.
2. **Optional:** In the **Computer Name** field, enter the name of the computer where the server you want to connect to is installed.
 - If you are configuring a full server or in a multiple server environment, enter the name of the computer that the server you want to connect to is installed on.
 - If you are configuring a server cluster environment, enter the cluster name used when you set up your cluster.
3. **Optional:** In the **Instance Name** field, enter the name of the server instance you want to connect to.
4. **Optional:** In the **HTTP Port** field, enter the port number required to allow communication with the server.
5. Click **Save**, and then click **Exit**.

Modify the Microsoft Active Directory Services

The **Configure AD LDS Integration** tool provides the ability to configure Active Directory Lightweight Directory Service for your production environment.

Procedure

1. On the server machine, click **Start > All Programs > General Electric > Workflow > Configuration > Configure Directory Services**.
The **AD LDS Integration** page appears.

2. To integrate Active Directory Lightweight Directory Services (AD LDS), do one of the following.

If AD LDS is...	Then...
NOT installed	You are given information regarding its use. If it is not required, then click Next .
Installed in a test and/or development environment	Select the Do not use Microsoft Directory Services check box, and then click Next .
Installed in a production environment	Click Next .

3. If you are using AD LDS, perform the following procedure:
- Clear the Do not use AD LDS check box.
 - In the **Directory Instance** section, in the **Name** field, enter an instance name for the ADAM directory.
 - In the **Port (LDAP)** and **Port (SSL)** fields, enter valid port numbers.
 - In the **User Credentials** section, in the **Name** field, enter the name of a user that is a member of the local computer's Administrators Group.
 - In the **Domain** field, enter the name of your company's domain.
 - In the **Password** field, enter the password for the user you entered in the **Name** field.
4. Click **Save**, and then click **Exit**.

Modify your application server database

The **Configure Database** tool provides the ability to make changes to your SQL Server database, as well as change the database you want to connect to.

Before you begin

You must install SQL Server before you install Workflow. For more information, see **Software Requirements** in the Workflow IPI documentation.

About this task

In order for Workflow to connect to Microsoft® SQL Server at runtime, you must configure SQL Server to enable the *db_owner* role for the applicable login account. For more information, see **Enable Connection between Workflow and SQL Server**.

The Workflow installation automatically selects the default SQL Server instance. For example, if SQL Server 2012 and SQL Server 2008 are both installed, SQL Server 2008 may be the default instance. In this scenario, if you want to use SQL Server 2012 instead, you must select this instance as the default as part of manually configuring your SQL Server settings. For instructions, see steps **2** and **3** of this procedure.

Procedure

1. Click Start > All Programs > General Electric > Workflow > Configuration > Configure Database.
The **Database Configuration** page appears.
2. Specify the SQL Server database settings for this application based on the configuration of your environment.

To configure SQL Server settings...	Then...
Automatically	<p>a. Select the Use a local database with Windows Authentication check box.</p> <div data-bbox="938 638 1422 1041" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: To use this option, the following SQL Server conditions apply:</p> <ul style="list-style-type: none"> ▪ Installed locally, AND ▪ Designated as default Local Host, AND ▪ Uses Windows Authentication </div> <p>b. Proceed to step 4.</p>
Manually	<p>a. Clear the Use a local database with Windows Authentication check box.</p> <p>b. Proceed to step 3.</p>

3. Specify your SQL Server settings, as follows:

- a. In the **Server** field, enter or select the name of the SQL Server that you want to connect to.

**Note:**

If the SQL Server is installed locally with a default instance, you may enter
localhost.

- b. In the **Database** field, enter the name of your SQL Server database, or click the drop-down arrow to search for all databases located on the specified server.

**Note:**

If the specified database does not exist, it will be created for you.

- c. From the **Authentication** list, select the type of authentication that you want to use, and then proceed based on your selection.
 - If you select Windows Authentication, proceed to step **4**.
 - If you select SQL Server Authentication, proceed to step **3.d**.
 - d. Enter the `User Name` and `Password` that are configured for SQL Server authentication, and then proceed to step **4**.
4. Click **Save**, and then click **Exit**.

Configure a server instance

The **Configure Server** tool is used to create an application server instance, as well as to configure server clustering.

Before you begin



Important:

The default user setting for this service is Local System Account. When upgrading your application, the user setting will return to the default setting.

About this task



Note:

If you are configuring servers for clustering, you must ensure that you run this tool on all nodes of the cluster.

Procedure

1. On the server machine, click **Start > All Programs > General Electric > Workflow > Configuration > Configure Server**.
2. Select the Create an Application Server Instance option.
3. In the **Name** field, enter a name for the server instance.
4. **Optional:** In the **Description** field, enter a description for the server instance.
5. Click the **Advanced Configuration** arrow.
6. Select or clear the **Do not create Windows Service for this Server Instance** check box, depending on whether you want this server instance to run as a Windows service.
7. If you are configuring the server instance to run as a Windows service, and the architecture of the computer you are configuring the instance on is 64-bit, configure the **Run service under the following architecture** setting to indicate your preference for operational mode as either 32-bit or 64-bit.

8. In the **Service Ports** fields, enter the port numbers for each of the service port types.
9. In the **P2P Configuration** section:
 - a. In the **IP Address** field, enter the IP address of the server instance.



Note:

Peer servers will contact this server instance at the specified IP address. If the specified address is 255.255.255.255, then the address will be determined at server start as the first IPv4 address found in the DNS records for the hosting machine's host name. If the address is incorrect, then an appropriate address should be specified here.

- b. In the **Port** field, enter the port number for the server instance. This value must be unique if multiple server instances are run on the same machine.
10. Click **Save**, and then click **Exit**.

Related reference

[Windows Services for Workflow \(on page 57\)](#)

Related information

[Server Clustering and Failover \(on page 70\)](#)

Configure product options

Using the **Configure Product Options** tool, you can install and configure product options that you license and/or service providers that you have created.

Before you begin

Proficy services must be stopped on the server prior to configuring product options.

If you have created and packaged your service provider, you must copy it to an accessible location.



Note:

Refer to **Architecture Considerations** for guidelines on deploying service providers based on architecture mode.



Important:

Product Options Installation Information. When using the ActiveX Task List, product option updates made on the server are not automatically deployed to applications using the Task List.



The ProficyUpdater.exe application must be run manually, with administrative privileges, in order to deploy production option updates on each client machine using ActiveX Task List.

Install one of the following server types:

Prerequisites

[Install a single application server and client \(on page 79\)](#)

[Install Multiple Servers \(on page 91\)](#)

Procedure

1. On the server machine, click **Start > All Programs > Proficy > Workflow > Configuration > Configure Product Options**.

The **Configure Product Options** tool appears, displaying the product options currently configured on the server, including those included by default as part of the server installation.

2. Proceed as applicable based on whether you are installing or uninstalling a product option:

If you are...	Then do this...
<p style="text-align: center;">Installing a product option.</p>	<p>a. Click Install.</p> <p>Your directory opens at the ProductOptions folder within the Workflow installation directory.</p> <p>b. Select the product option file you want to install, and then click Open.</p>
<p style="text-align: center;">Uninstalling a product option.</p> <div data-bbox="277 1350 841 1570" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  <p style="text-align: center;">Note:</p> <p style="text-align: center;">You cannot uninstall the default product options that are included in the server installation.</p> </div>	<p>a. Select the product option you want to uninstall.</p> <p>b. Click Uninstall.</p> <p>c. Click Save, then click Exit.</p>

3. Select the product option file you want to install, and then click **Open**.



Note:

You can select only one product option at a time to install.

The product option is installed, and then the **Configure Site** dialog box appears.

4. In the **Application Server Instances and Enabled Product Options** pane, select the server that you want to host the product option.
5. In the **Disabled Product Options** pane, click the listing for the product option, and then do one of the following:
 - Click **Enable**.
 - Drag and drop the listing to the **Application Server Instances and Enabled Product Options** pane.
6. Indicate whether the product option is essential by selecting or clearing the Essential check box.
7. Click **Save**, and then click **Exit**.

Related information

[Architecture Considerations \(on page 54\)](#)

Change architecture mode

When initially installing an application server or extension server (that is, Workflow or User) on a 64-bit computer, you can elect to run the server in either 32-bit or 64-bit mode. If your environment changes subsequent to installation, you can use the **Configure Server** tool to change the architecture mode accordingly.

Prerequisites

[Architecture Considerations \(on page 54\)](#)

Procedure

1. On the machine that the server is installed on, click **Start > All Programs > General Electric > Workflow > Configuration > Configure Server**.
The **Configure Server** tool appears.
2. Select the Create an Application Server Instance option.
3. Click the **Advanced Configuration** arrow.
4. In the **Run service under the following architecture** section, select 32-bit or 64-bit to indicate the operational mode you want to switch to.
5. Click **Save**, and click **Exit**.

Add or remove performance counters

Using this procedure, you can manually add performance counters to your system, or remove them. You must have administrative privileges in order to perform this action.

Procedure

1. From the following location, you can add or remove performance counters: `<installfolder>/ProficyServer.exe`.
2. In `ProficyServer.exe`, enter the applicable command line.

To...	Enter...
Add counters	<code><installfolder>ProficyServer[exe] / installperformancounters</code>
Remove counters	<code><installfolder>ProficyServer[exe] / uninstallperformancounters</code>

**Important:**

If the counters are added/deleted (installed/uninstalled) after the server is started, then the server and the performance monitoring tool (perfmon) must be re-started for this change to take effect.

Related information

[View performance counters \(on page 78\)](#)

Display the host server name in the Workflow client

By default, the server name is displayed in the client, next to the server state icon. However, if you want to display a more meaningful name for your server, you can configure an alias so that the alias is displayed in the client rather than the actual server name.

Before you begin

You must install Workflow 2.5 SIM3 (or greater) before performing this task.

Procedure

1. On the Workflow client computer, navigate to the `C:\Program Files (x86)\Proficy\Proficy Workflow\Program\Data` folder.
2. Open the `UserStartup.dita` file in a text editor.

The following code shows samples of multiple server names and aliases. You can copy this code into the `UserStartup.dita` file, before the closing `</ServerConnectionInfo>` element, and then change the `Name` and `Alias` elements to reflect your server information.

```
<ServerAliases>
  <ServerAlias>
```

```

<Name>localhost</Name>

<Alias>QA Server</Alias>

</ServerAlias>

<ServerAlias>

<Name>Product-WIN81</Name>

<Alias>Development Server</Alias>

</ServerAlias>

<ServerAlias>

<Name>devwin7ex64dev</Name>

<Alias>Product Server</Alias>

</ServerAlias>

</ServerAliases>

```

When you launch the Workflow client, the alias that you configured for the server is displayed next to the state icon.

Client and Server Configurations

Workflow supports adding server and client configurations that are not included in the respective configuration files.



CAUTION:

Performance can be affected if values are adjusted in the configuration files.

The following steps apply to both client and server configuration files.

1. Determine which configuration(s) and value(s) you want to add to the applicable file.
2. Use the following XML element example to configure the file.



Note:

All new configurations must be added to the `<appSettings>` element.

```

<appSettings>

  <add key="ServerStatePingInterval" value="10" />

</appSettings>

```

**Note:**

- If the configuration is for a client, then start a new client for the new configuration to take effect.
- If the configuration is for the SOA or STS service, then restart that service for the new configuration to take effect.

Client Configuration

All client configurations are added to the `ProficyClient.exe.config` file.

Configuration	Description
ServerStatePingInterval	Specifies the server state refresh interval. Sites with a large number of clients may need to increase this interval to prevent high load from the continuous pings. The default value is 30 seconds.
NavigatorSearchMaximumDisplayResults	Specifies the maximum number of objects that can be displayed in the navigator's search results area. The default value is 500.
ChannelPoolMaxIdleAgeHours	Specifies the maximum amount of time (in hours) that a channel is held idle in the pool before being destroyed due to inactivity. The default value is 24 hours; the minimum value is 1 hour. If a valid value is not found or the value is less than the minimum, then the default value is used.

**Note:**

GE recommends a value no greater than 168 hours (one week) for practicality and to avoid possible overflow when converting to milliseconds. It is significant because channel creation is expensive when the server is busy. Thus we want to reuse channels where possible, especially where sites have an expensive event or operation occurring at a regular frequency. Set-

Configuration	Description
	 ting this value to a period longer than the site event frequency allows channels to be maintained from the last occurrence so that few, if any, channels need to be created to satisfy the operation.
OperationTimeoutSeconds	Specifies the WCF operation timeout. The default value is 600 seconds.
ReceiveTimeoutMinutes	Specifies the WCF receive timeout. The default value is 48 hours. If this value is configured as 0, the TimeSpan maximum value is used.  Note: The time span value must be at least double the ChannelPoolMaxIdleAgeHours value.
UseDefaultWebProxy	Indicates whether the auto-configured HTTP proxy of the system should be used, if available. The default value is True.
UserPermissionsRefreshMinutes	Specifies the length of time it takes for a permission change to take effect. The default value is 1 minute.

Server Configuration: SOA Service

The following configurations are added to the ProficiencyServer.exe.config file.

Configuration	Description
WorkflowDomainServiceStartupPollingFrequency-InSeconds	Specifies the Workflow domain service start up polling frequency (in seconds). This configuration is used to check the essential services status. Waits for all essential service providers to start before starting the Workflow domain service. The de-

Configuration	Description
	fault value is 10 seconds. If the configured value is less than 1 second, then the default value is used.
CacheBasePath	Specifies the root directory for all client caches. The default value is C:\Users\ <myuser-name>\AppData\Local.</myuser-name>
LicensePollPeriodMinutes	Specifies the read and update license interval. The default value is 10 minutes. The maximum value is 24*60 minutes and the minimum value is 1 minute.
sqlCommandTimeout	Specifies the SQL command timeout (in seconds), from the configure file. A value of 0 indicates no limit (an attempt to execute a command will wait indefinitely). The default value is 30 seconds.
ChannelPoolMaxIdleAgeHours	<p>Specifies the maximum amount of time (in hours) that a channel is held idle in the pool before being destroyed due to inactivity. The default value is 24 hours; the minimum value is 1 hour. If a valid value is not found or the value is less than the minimum, then the default value is used.</p> <div data-bbox="824 1171 1422 1890" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: GE recommends a value no greater than 168 hours (one week) for practicality and to avoid possible overflow when converting to milliseconds. It is significant because channel creation is expensive when the server is busy. Thus we want to reuse channels where possible, especially where sites have an expensive event or operation occurring at a regular frequency. Setting this value to a period longer than the site event frequency allows channels to be maintained from the last occurrence so that few, if any, channels need to be created to satisfy the operation.</p> </div>

Configuration	Description
logStatistics	<p>Indicates whether to log client dispatch details. The default value is <code>False</code>.</p> <div data-bbox="824 390 1422 562" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: This configuration must enable tracing in the configuration file:</p> </div> <pre data-bbox="836 594 1401 720" style="background-color: #F0F0F0; padding: 5px;"> <logger name="ProxyClientDispatcher" threshold="DEBUG"> <file value=%TraceDir%\ProxyClientDispatcher.log"/> </logger> </pre>
ReceiveTimeoutMinutes	<p>Specifies the WCF receive timeout. The default value is 48 hours. If this value is configured as 0, the <code>TimeSpan</code> maximum value is used.</p> <div data-bbox="824 915 1422 1129" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: The time span value must be at least double the <code>ChannelPoolMaxIdleAgeHours</code> value.</p> </div>
OperationTimeoutSeconds	<p>Specifies the WCF operation timeout. The default value is 600 seconds.</p>
UseEquipmentClassStoredProcedures	<p>Indicates whether to use stored procedures to update equipment class properties to improve performance. The default value is <code>False</code>.</p>
HistorianDeleteTimeout	<p>Specifies the Historian deletion timeout. The default value is 300 seconds.</p>
HistorianManualSyncOnly	<p>Indicates whether Historian synchronization will only be triggered manually rather than automatically on startup. The default value is <code>False</code>.</p>
HistorianCustomCollectorNames	<p>Specifies the Historian collector names to synchronize that are not registered collectors in your Historian server. Use a comma (,) as a separator.</p>

Configuration	Description
EquipmentProvisioningTimeout	<p>This timeout configuration is for the equipment provisioning tool to process provisioning commands. By default, the equipment provisioning process times out if the processing time exceeds 10 minutes. For very large amounts of commands (for example, tens of thousands), 10 minutes may be insufficient preventing the provisioning from completing. If required, the timeouts can be adjusted.</p> <p>The following appSetting can be added. The value is the timeout in seconds.</p> <pre data-bbox="818 814 1382 888"><add key="EquipmentProvisioningTimeout" value="1200" /></pre> <p>If a value greater than 10 minutes is used, the WCF timeout must also be adjusted higher than the default of 10 minutes.</p> <pre data-bbox="818 1083 1317 1157"><add key="OperationTimeoutSeconds" value="1200" /></pre>
DisplayCacheFolder	<p>Specifies a custom folder path to store user display projects and assemblies on client machines. The path must be absolute; however, environment variables can be used as part of the path. For example:</p> <pre data-bbox="818 1381 1344 1413">"%ProgramData%\Workflow Displays"</pre> <p>The default path is the Proficy Workflow folder in your My Documents folder.</p>

Server Configuration: SOASecurity Token Service

The following configurations are added to the ProficySTS.exe.config file.

Configuration	Description
MaximumClockSkew	Specifies the maximum allowable time difference between the system clocks of the two parties that

Configuration	Description
	are communicating. The default value is 5 minutes. Use a maximum value of 10 minutes.

Server Limit Overrides

There are many limits in the Proficy SOA server that can be changed without recompiling any source. These limits are changed by adding a named file with the new limits to the Program folder.

The following steps apply to all server limits that you want to override.

1. Determine which server limit(s) and limit group(s) you want to override.
2. In the Program folder, (that is, <install dir>\Proficy\Proficy Workflow\Program, create a file called <LimitGroup>.lim. For example, to change a limit in the SystemLimits group, name the file SystemLimits.lim.
3. Add the XML content to the file, as shown in the following example. Add a <limit> element for each limit you want to override.

```
<?xml version="1.0" encoding="utf-8" ?>
<limitOverrides>
  <limit name="QueryResultLimit" warningThreshold="5000" errorThreshold="5000"/>
</limitOverrides>
```

DmsLimits

Limit	Description
ConstraintLimit	Specifies the maximum number of references between the same parent and child. The default warningThreshold is 4 and the default errorThreshold is 4.
PrimaryKeyLimit	Specifies the maximum number of components (properties and references) that can make up a primary key. The default warningThreshold is 6 and the default errorThreshold is 6.
AttributeLimit	Specifies the maximum number of attributes that can be defined in a Model Object Type. The default warningThreshold is 100 and the default errorThreshold is 100.

Limit	Description
ReferenceLimit	Specifies the maximum number of references that can be defined on a Model Object Type. The default warningThreshold is 20 and the default errorThreshold is 20.
SessionObjectLimit	Specifies the maximum number of objects that a session can contain. The default warningThreshold is 20000 and the default errorThreshold is 20000.
ClassesPerInstanceLimit	Specifies the maximum number of Classes that can be added to an instance. The default warningThreshold is 50 and the default errorThreshold is 50.
ClassPropertyLimit	Specifies the maximum number of properties that can be added to a class. The default warningThreshold is 1000 and the default errorThreshold is 1000.
TotalPropertyLimit	Specifies the maximum number of properties (that is, the associated class properties and instance properties combined) that can be on an instance. The default warningThreshold is 5000 and the default errorThreshold is 5000.

ProductionLimits

Limit	Description
S95IdLength	Specifies the maximum length of the S95 ID attribute for S95 resources. The default warningThreshold is 50 and the default errorThreshold is 50.
DescriptionLength	Specifies the maximum length of the Description attribute for S95 resources. The default warningThreshold is 255 and the default errorThreshold is 255.
UnitOfMeasureLength	Specifies the maximum length of the unit of measure fields for S95 resources. The default warning-

Limit	Description
	Threshold is 50 and the default errorThreshold is 50.

ProductionModelLimits

Limit	Description
SegmentHierarchyDepth	Specifies the maximum number of levels that can be in the process or product segment hierarchy. The default warningThreshold is 4 and the default errorThreshold is 4.
Number OfProcessSegments	Specifies the maximum number of process segments that can be in a project. The default warningThreshold is 5000 and the default errorThreshold is 5000.
NumberOfAssociatedProcessSegments	Specifies the maximum number of process segments that can be associated with a product segment. The default warningThreshold is 1 and the default errorThreshold is 1.
NumberOfResourceSpecificationsPerSegment	Specifies the maximum number of a particular type of resource specification (material/equipment/personnel) that can be created for one segment. The default warningThreshold is 500 and the default errorThreshold is 500.
NumberofResourceSpecificationsPerSegmentRequirement	Specifies the maximum number of a particular type of resource specification (material/equipment/personnel) that can be associated with one segment requirement. The default warningThreshold is 1000 and the default errorThreshold is 1000.
NumberOfResourceSpecificationsForMasterSegment	Specifies the maximum number of a particular type of resource specification (material/equipment/personnel) that can be created for a Master Segment. The default warningThreshold is 1200 and the default errorThreshold is 1200.

Limit	Description
NmberOfSegmentsPerWorkDefinition	Specifies the maximum number of segments in a work definition. The default warningThreshold is 50 and the default errorThreshold is 50.
NumberOfOverallSegmentsPerWorkDefinition	Specifies the maximum number of segments, under all hierarchy levels, in a work definition. The default warningThreshold is 200 and the default errorThreshold is 200.
NumberOfParametersPerSegmentRequirement	Specifies the maximum number of parameters that can be associated with a segment requirement. This value is twice the maximum number of parameters per process/product segment. The default warningThreshold is 200 and the default errorThreshold is 200.
NumberOfWorkRequestsPerArchive	Specifies the maximum number of work requests that can be archived at once. The default warningThreshold is 500 and the default errorThreshold is 500.

SystemLimits

Limit	Description
ExampleLimit	Specifies an example of a Server Core Limit. The default warningThreshold is 123 and the default errorThreshold is 124.
NameLength	Specifies the default system limit for the maximum length of object names. The default warningThreshold is 50 and the default errorThreshold is 50.
PropertyNameLength	Specifies the default system limit for the maximum length of object property names. The default warningThreshold is 200 and the default errorThreshold is 200.
DescriptionLength	Specifies the default system limit for the maximum length of object descriptions. The default warning-

Limit	Description
	Threshold is 255 and the default errorThreshold is 255.
QueryResultLimit	Specifies the maximum number of results to return in a WCF query. The default warningThreshold is 5000 and the default errorThreshold is 5000.

WorkflowLimits

Limit	Description
CategoryDefinitionsLimit	Specifies the maximum number of category definitions. The default warningThreshold is 1000 and the default errorThreshold is 1000.
CompletionCodeDefinitionsLimit	Specifies the maximum number of completion code definitions. The default warningThreshold is 1000 and the default errorThreshold is 1000.
ConcurrentDebugSessionsLimit	Specifies the maximum number of concurrent debug sessions. The default warningThreshold is 10 and the default errorThreshold is 10.
ConcurrentWorkflowInstancesLimit	Specifies the maximum number of concurrent workflow instances. The default warningThreshold is 1000 and the default errorThreshold is 1000.
EventScopeEventHandlerLimit	Specifies the maximum number of events allowed to be queued for an event scope activity at runtime. The default warningThreshold is 10 and the default errorThreshold is 10.
FaultDefinitionLimit	Specifies the maximum number of fault definitions. The default warningThreshold is 1000 and the default errorThreshold is 1000.
ResourceUserVersionLimit	Specifies the maximum number of resource user versions. The default warningThreshold is 1000 and the default errorThreshold is 1000.

Limit	Description
SubprocessDefinitionLimit	Specifies the maximum number of subprocess definitions. The default warningThreshold is 1000 and the default errorThreshold is 1000.
TaskClientTaskHistoryResultsLimit	Specifies the maximum number of task client and history results. The default warningThreshold is 1000 and the default errorThreshold is 1000.
WorkflowDefinitionsLimit	Specifies the maximum number of workflow definitions. The default warningThreshold is 1000 and the default errorThreshold is 1000.
WorkflowScheduleLimit	Specifies the maximum number of workflow schedules. The default warningThreshold is 3000 and the default errorThreshold is 3000.
UserActivityLimit	Specifies the maximum number of user activities. The default warningThreshold is 500 and the default errorThreshold is 500.
DataItemLimit	Specifies the maximum number of data items. The default warningThreshold is 100 and the default errorThreshold is 100.
UserActivityNestingLimit	Specifies the maximum number of user activity nesting levels. The default warningThreshold is 5 and the default errorThreshold is 5.

FormLimits

Limit	Description
EventDefinitionsLimit	Specifies the number of event definitions allowed to be defined on a form. The default warningThreshold is 15 and the default errorThreshold is 15.
DataItemDefinitionsLimit	Specifies the number of data item definitions allowed to be defined on a form. The default warningThreshold is 20 and the default errorThreshold is 20.

Limit	Description
InputParameterDefinitionsLimit	Specifies the number of input parameter definitions allowed to be defined on a form. The default warningThreshold is 100 and the default errorThreshold is 100.
OutputParameterDefinitiosLimit	Specifies the number of output parameter definitions allowed to be defined on a form. The default warningThreshold is 100 and the default errorThreshold is 100.
ServiceProviderDefinitionsLimit	Specifies the number of Service Provider method definitions allowed to be defined on a form. The default warningThreshold is 20 and the default errorThreshold is 20.

ResourceLimits - For Database Provider

Limit	Description
MaximumDatabaseConnectionLimit	Specifies the maximum number of connections that can be held. The default warningThreshold is 100 and the default errorThreshold is 100.
MaximumDatabaseStatementsLimit	Specifies the maximum number of statements that can be held. The default warningThreshold is 500 and the default errorThreshold is 500.

ResourceLimits - For Web Service Provider

Limit	Description
MaximumWebServicesLimit	Specifies the maximum number of web services that can be held. The default warningThreshold is 100 and the default errorThreshold is 100.

Upgrade Workflow

Upgrading Workflow

Please refer to the WorkflowIPI PDF for information on upgrading Workflow, compatibility with earlier versions and other GE Products. Read this topic for general information about upgrading your applications.



Important:

When you upgrade the application server, you must also upgrade all of the remote clients that connect to that server to use the same version of Workflow.

Upgrade Paths

You can upgrade Workflow without having to uninstall the program.

- From Workflow 2.5 SP4 to Workflow 2.6
- From Workflow 2.5 SP3 to Workflow 2.5 SP4
- From Workflow 2.2 SP2 to Workflow 2.5 SP4
- From Workflow 2.5 SP1 to Workflow 2.5 SP4
- From Workflow 2.5 to Workflow 2.5 SP4
- From Workflow 2.5 SP2 to Workflow 2.5 SP3
- From Workflow 2.5 SP1 to Workflow 2.5 SP3
- From Workflow 2.5 to Workflow 2.5 SP3
- From Workflow 2.2 SP1 to Workflow 2.5 SP3
- From Workflow 2.5 SP1 to Workflow 2.5 SP2
- From Workflow 2.5 to Workflow 2.5 SP2
- From Workflow 2.5 to Workflow 2.5 SP1
- From Workflow 2.1 to Workflow 2.5
- From Workflow 2.2 service pack 1 (SP1) to Workflow 2.5
- From Vision 6.2 to Vision 6.3



Note:

All versions of Vision prior to version 6.3 must be uninstalled before being upgraded to Workflow 2.5 SP1.

- From Workflow 2.1 to Workflow 2.2 service pack 1 (SP1)
- From Workflow 2.0 to Workflow 2.2 service pack 1 (SP1)

- From Workflow 1.5 service pack 4 (SP4) to Workflow 2.2 SP1



Note:

You must first upgrade Workflow 1.5 SP4 to Workflow 2.0.

If you are upgrading from an earlier version of Workflow, contact Technical Support.

GE Historian

If you use GE Historian as a data source, then when you upgrade Workflow to v2.5 SP2 the existing GE Historian server address is automatically displayed in the **Server Name** field in the **Data Source Editor**. You cannot change the server's alias (that is the name displayed in the **Historian Servers** list in the navigator). To change the alias, you must delete the connection, and then add a new one. For more information, see **Change the Historian Server** and **Add an Historian Server**.

Legacy Certificates

Starting with Workflow 2.5, legacy certificates are no longer supported.

Upgrading Extension Servers

If you are installing or upgrading an extension server (Workflow or User), you must have already installed or upgraded an application server of the same version to host the Core services on a separate machine

Upgrading Microsoft® .NET Framework 4.5 (Full Framework)



Note:

Custom display or form assemblies that target earlier versions of .NET will continue to function as before. However, saving changes made to such custom assemblies requires that they be upgraded, which is accomplished in different ways, based on the version of .NET:

- For an assembly that pre-dates .NET 4.0, you must upgrade the assembly when you open it. Failure to do so results in an error and the inability to save the edited assembly.
- For an assembly that targets .NET 4.0, the assembly is upgraded automatically when opened, and changes to such assemblies can be saved as before.

Service providers that target versions of the .NET framework before version 4.5 must be recompiled targeting .NET 4.5.

Upgrading Web-based Forms for the Task List

If you have web-based forms as part of an existing workflow that you have created prior to the release of Workflow 2.2 SP1, you must bind them again inside the Form activity (see **Configure a Form Activity**) in order for them to function in the Task List.

**Note:**

This upgrading issue affects existing web-based forms only, and not HTML5 web forms that are created in Workflow 2.2 SP1.

Upgrading a Clustered Environment

After upgrading Workflow on the primary server machine, you must reconfigure that machine as the primary server (see **Configure the Primary Cluster Server**) in the cluster, as well as reconfigure the failover server (see **Configure the Failover Cluster Server**) in the cluster.

SQL Server Database Replication

When upgrading the application, if you are using a replicated database and have replication turned on, you must turn it off, upgrade, and then turn it on again. You must also upgrade the replicated database. For more information, see **Upgrade a replicated database**.

Password Security

During an upgrade installation, the account lockout capability is automatically enabled; all other password security features are disabled, by default. To enable any of the other password security features, you must use the Configure Security tool after the upgrade installation has successfully completed.

Windows Users

Workflow 2.0 supports Windows domain names with personnel names and login names. If you choose to use the new functionality included with Workflow 2.0 (that is, active directory universal or global groups mapped to Workflow groups), then when your Windows users log in, their personnel name and login name will both be updated to include the domain name, and the personnel name will change to match the login name. For example, the personnel name *John Smith* and login name *johnsmith* will both change to *<domain name>\johnsmith*.

**CAUTION:**

If you were using Windows user accounts in a prior version of Workflow, any workflows that reference individual users will no longer work! These personnel names will NOT be updated when



the Windows users log in. You must manually change these names to include the domain name with a backslash character between the domain name and personnel name. However, if you reference personnel classes in your workflows, no change is required; the workflows will work as they did in the previous version of Workflow.

Install a Reporting Database

Reporting Database

Workflow includes an option to install a reporting database. This database can be used with any valid reporting tool, and allows you to build reports on up-to-date data that you synchronize from your production database.

**Note:**

The user account used for reporting synchronization (this can be a different user than the SOA database user), must have the following SQL server roles:

- *SQLAgentOperatorRole* for the msdb database, in order to initiate the SQL Agent job
- *Db_owner* on the reporting database

For more information on the Reporting Database, see **Reporting**.

Related information

[Reporting \(on page 662\)](#)

Install the reporting database

The reporting database allows you to use the data synchronized from your production database to build reports for your facility. You can use any reporting tool to extract the information from the reporting database and create your reports.

Before you begin

You must have installed a supported SQL Serve version before you install the reporting database (see **Software Requirements**). In addition, you must select the SQL Server Integration Service (SSIS) component when you install SQL Server. Select the SQL Reporting Services component, as well, if you want to use SQL as your reporting tool.

**Note:**

We recommend that you install the reporting database on a separate computer from the production database to prevent performance issues on the production database. In this scenario, additional configuration changes are required. Refer to **Configure Component Services** for more information. However, if the application server computer meets the requirements for both databases, they can be installed on the same computer.

About this task

If more than one version of SQL Server is installed, the integration services used for the reporting database will be the most recent version.

**Note:**

If you plan on frequent synchronization for reporting purposes, we recommend that you replicate your database and run reporting against the replicated database. This measure reduces the risk of system timeout or shutdown due to locking of tables in the production database. If, upon trying to save data after replication, you receive an error similar to "System.Data.SqlClient.SqlException (0x80131904): Length of LOB data (88064) to be replicated exceeds configured maximum 65536," use SQL statements to increase the volume of data to replicate.

Example:

```
sp_configure 'max text repl size', 2147483647  
  
GO  
  
RECONFIGURE  
  
GO
```

For information on setting up database replication, visit the Microsoft Developer Network website and review the article, [SQL ServerReplication](#).

Procedure

1. From the application splash screen, click **Install Reporting Database**.

**Note:**

If this splash screen does not appear, run `SetupReporting.exe` on the root directory of the installation folders.

The **License Agreement** page appears.

2. Review the license agreement, and then click **I Agree**.

The **Installation Folder** page appears.

3. Accept the default destination folder or browse for a new location, and then click **Next**.
4. In the **Configure Reporting SQL Server** area, enter the following information, and then click **Next**.

- a. In the **Server** field, accept the default value if the reporting database is on the local machine with the default instance. If the SQL Server named instance is used, enter the SQL Server name and instance name; for example, `<servername>\<instance name>`.



Note:

The reporting SQL Server must be local.

- b. In the **Database** field, enter the name of the reporting database. This creates a new database; however, if a database of the same name already exists, this database will append to the existing one.
 - c. From the **Authentication** list, select the type of authentication you want to use. If you choose to use integrated security, select **Windows Authentication**; otherwise, select **SQL Server Authentication**, and then enter the user name and password for the SQL Server.
5. In the **Configure SQL Server** area, perform one of the following actions:
 - Leave the **Use a local SQL database with Windows Authentication** check box selected, and then click **Next**.
 - Clear the **Use a local SQL database with Windows Authentication** check box, enter the following information, and then click **Next**.
 - a. In the **SQL** field, enter the name of the SQL Server where the SQL database is located.
 - b. In the **Database** field, enter the name of the production (SQL) database that you will connect to for synchronization.
 - c. From the **Authentication** list, select the type of authentication you want to use. If you choose to use integrated security, select **Windows Authentication**; otherwise, select **SQL Server Authentication**, and then enter the user name and password for the SQL Server.
 6. In the **Workflow Server Configuration** area, enter the following information, and then click **Next**.
 - In the **Server computer name** field, enter the fully qualified name of the server computer.
 7. In the **Configure Security** area, enter an administrator user name and password.

 **Important:**

- You cannot use a Windows user name and password to configure security for the reporting database; it must be a valid Windows administrator user name and password.
- The **Advanced user authentication settings (Use SSO, Allow Password Change, Enforce User Lockout, and Enforce Password Complex Rules)** do not apply when installing the reporting database.

8. Click **Install**.

9. When the installation is complete, click **Exit**.

Related information

[Reporting Database \(on page 164\)](#)

[Manually synchronize databases \(on page 690\)](#)

[Upgrade a replicated database \(on page 169\)](#)

[Configure Component Services \(on page 167\)](#)

Configure Component Services

If your production database is on a different computer than the Reporting Database, the following configuration changes are required.

Before you begin

 **Important:**

You can perform these steps either before or after installing the Reporting Database; however, you must complete these configuration changes before you run your reporting data synchronization.

 **Note:**

For this procedure, your production database server is the "server," while the reporting database server is the "client."

Procedure

1. Verify that the Distributed Transaction Coordinator service is running on both the server and client computers.
 - a. From the **Start** menu, point to **Administrative Tools**, and then click **Services**.
 - b. If the Distributed Transaction Coordinator service is not running, right-click it, and then click **Start**.
2. On the server computer, from the **Start** menu, point to **Administrative Tools**, and then click **Component Services**.
3. In the left navigation tree, click **Component Services**, and then expand **Computers**.
4. Right-click **My Computer** and select **Properties**.

The **My Computer Properties** dialog box appears.
5. Click the **MS DTC** tab.



Important:

If you are using Windows 7, Windows Server 2008, or Windows Server 2012, you must use the following path to retrieve these computer properties: Component Services>Computers>My Computer>Distributed Transaction Coordinator>Local DTC.

6. Click **Security Configuration**.

The **Security** page appears.
7. Select the following check boxes:
 - **Network DTC Access**
 - **Allow Remote Clients**
 - **Allow Inbound/Outbound Administration**
 - **Enable Transaction Internet Protocol (TIP) Transactions**
8. Click **OK**.

A message appears, stating that "MS DTC Service will be stopped and restarted. All dependent services will be stopped. Please press Yes to proceed." Click **Yes**.
9. In the **My Computer Properties** dialog box, click **OK**.
10. If required, reboot your production database server.



Note:

We recommend that you reboot your production database server.

11. On the client computer, repeat steps 2-6.
12. Select the **Network DTC Access** and **Allow Inbound/Outbound Administration** check boxes.

The DTC Service is stopped and restarted.

13. Restart the client computer.
14. Verify that the Distributed Transaction Coordinator service is running on both the server and client computers, and, if required, repeat step 1.

Uninstall the reporting database

When you uninstall reporting, the database remains intact; however, you lose the ability to synchronize the data from your production database. Uninstalling reporting does not affect the production database or the data contained in it; you can still run reports on the data stored in the database.

Procedure

1. From the **Control Panel**, click **Programs and Features**.

The **Uninstall or change a program** dialog box appears, displaying all of the programs installed on the computer.

2. Select **Reporting**, and then click **Uninstall**.
3. In the confirmation message box, click **Yes**.

The **Uninstall Reporting** message box appears stating, "Uninstalling the Proficy Reporting application will not remove the Reporting database or the SQL Agent job from SQL Server. For a complete uninstall of this application, these should be removed manually following the uninstall."

4. Click **OK**.

Upgrade a replicated database

When upgrading the application, if you are using a replicated database and have replication turned on, you must turn it off, upgrade, and then turn it on again. You must also upgrade the replicated database.

Procedure

1. Disable any scheduled synchronization jobs for the Reporting Database.
2. Open the **Services** window to stop all services: click **Start > Control Panel > Administrative Tools > Services**.
3. In Microsoft SQL Server Management Studio, generate the create and drop scripts:



Note:

For use later in this procedure, you must note the location of where you saved the scripts.

- a. In the Subscriber/Distributor instance, right-click the **Replication** folder, and then click **Generate Scripts**.

The **Generate SQL Script** dialog box appears.

- b. To generate the **Create** script, in the **Script the commands** section, select **To create or enable the components**, and then click **Generate Script**.
 - c. To generate the *Drop* script, in the **Script the commands** section, select **To drop or disable the components**, and then click **Generate Script**.
4. To disable replication, execute the *Drop* script.
5. Verify that all objects beneath the Microsoft SQL Server Management Studio **Replication** folder have been deleted.
6. From the installation directory where your SQL Server is hosted, go to `Program Files \Proficy\Proficy Workflow\Program`, and then run `ProficyInstaller` to upgrade the `ProficyServer` programs and database.
7. Edit the *Create Replication* script.
 - a. In Microsoft SQL Server Management Studio, from the **File** menu, click **File > Open File**.
 - b. Navigate to the folder where you saved the scripts generated in step 3.
 - c. Select the `Create SQL` script.
 - d. Locate the section denoted by this comment, `***** End: Script to be run at Publisher *****`, and then add the following lines after it.

**Note:**

You must change the name of the Distribution database and the user names to match your system.

```
-- Create users in the Distribution database
USE [DistributionDB];

GO

CREATE USER [QASYSTESTSQLSRV\SqlDistAgent]
  FOR LOGIN [QASYSTESTSQLSRV\SqlDistAgent]

EXEC sp_addrolemember N'db_owner',
  N'QASYSTESTSQLSRV\SqlDistAgent';

CREATE USER [QASYSTESTSQLSRV\SqlLogReaderAgent]
  FOR LOGIN [QASYSTESTSQLSRV\SqlLogReaderAgent];

EXEC sp_addrolemember N'db_owner',
  N'QASYSTESTSQLSRV\SqlLogReaderAgent';

CREATE USER [QASYSTESTSQLSRV\SqlSnapshotAgent]
```

```
FOR LOGIN [QASYSTESTSQLSRV\SqlSnapshotAgent];

EXEC sp_addrolemember N'db_owner',

N'QASYSTESTSQLSRV\SqlSnapshotAgent';
```

8. Add all of the passwords back into the file. The Create Generation process does not write the user account passwords to the file.

- a. Find all occurrences of `@job_password = null`.
- b. Replace `null` with the log on password that precedes it. If the preceding `@job_login` is also `null`, do not edit the password so that it remains set to `null`.

```
-- Before:

exec [SystemTest2v2].sys.sp_addlogreader_agent @job_login = N'QASYSTESTSQLSRV\SqlLogReaderAgent',

@job_password = null

-- After:

exec [SystemTest2v2].sys.sp_addlogreader_agent @job_login = N'QASYSTESTSQLSRV\SqlLogReaderAgent',

@job_password = 'proficy'

-- Do not change passwords for null job_logins

exec [SystemTest2v2].sys.sp_addqreader_agent @job_login = null, @job_password = null
```

9. Execute the **Create Replication** script.



Note:

This script will have some errors because it will try to publish database objects that were removed from the database as part of the upgrade.

10. Re-enable the replication, and then add the new database tables and objects to the publication in order to push them to the replicated database:

- a. Expand your database directory and the **Replication** folder, right-click **Local Publications**, and then select **Properties**.
- b. Select **Articles**, and then select all unchecked items to include them in the publication.



Note:

The subscription must be re-initialized to include the new articles. We recommend re-initializing all articles in the subscription using a full snapshot. For instructions, read the Microsoft Developer Network article, [Reinitializing a Subscription](#)

11. After replication is working against the upgraded database, install the Reporting upgrade, and point it to the replicated database.
12. Test reporting synchronization by running a manual Sync from the Workflow client.

Terminal Server

Remote Desktop Session Host

This information is intended for system integrators, IT administrators, and engineers responsible for setting up and optimizing a Workflow Remote Desktop Session Host environment.

This information assumes familiarity with your network environment, as well as:

- Microsoft® Windows® Server 2008 R2
- Microsoft® Windows® Server 2012
- Microsoft® Windows Licensing
- Microsoft® Windows Remote Desktop Services
- Citrix® technologies including licensing

Remote Desktop Services (Terminal Services)

With Remote Desktop Services for Microsoft® Windows® Server 2008 R2 and Windows® Server 2012, you can centrally manage and execute Workflow.

The Remote Desktop Session Host environment is a thin-client architecture where all application processing occurs centrally on the server. By installing a small piece of thin-client software from Microsoft® or connecting through an Internet Explorer 7 (or higher) browser, thin clients can initiate and run individual instances of Workflow on the server. Only graphic, keyboard, and mouse instructions are sent back and forth between the client and the server, which minimizes network traffic.

Using Remote Desktop Services provides:

Ease of maintenance

You install only one copy of Workflow on a server, from which multiple users can run clients. Upgrades and SIMs need to be installed only on the server.

Remote access with built-in technology

With Remote Desktop Communication (RDC) and Internet Explorer 7 or higher, clients from Windows 7 can connect to the Workflow Remote Desktop Session Host to access Workflow.

Security

Encryption and other security measures protect the data exchanges between the Workflow Remote Desktop Session Host and the clients.

Lightweight client machines

The Workflow Remote Desktop Session Host locally processes the software that the clients execute. Clients connecting to the server through a remote desktop session do not need the processing power usually required to run Workflow.

Specialized environments

Remote Desktop Services allows you to tightly control user accounts. For example, you can configure user accounts to start and execute a single program (Workflow). Workflow automatically starts at log in, and the users do not have access to the Windows desktop. When they exit Workflow, they log out of the Remote Desktop Session Host account.

Handheld environments

Remote Desktop Services enables wireless handheld devices to display Workflow.

Controlled access to files

With Windows file protection, you can limit the directories users are allowed to access and modify, such as for forms, displays, and import/export files.

Workflow Remote Desktop Session Host Environment

Workflow Remote Desktop Session Host allows multiple clients to run individual instances of Workflow from one server.

Thin clients access the Remote Desktop Session Host through Microsoft[®] Remote Desktop Protocol (RDP) or Citrix[®] Independent Computing Architecture (ICA) protocol. No Workflow software is installed or runs on the thin-client machine.

A separate session of Workflow runs on the Remote Desktop Session Host for each thin client. This allows very thin clients with minimal client-side resources to execute an individual instance of Workflow. The users' experience is nearly identical to running Workflow on their local machines. If you have clients and servers in your Workflow Remote Desktop Session Host environment, you can access and manage any of the servers from a thin client.

Getting Started with Remote Desktop Session Host

Before installing Workflow on a Windows[®] Server 2008 R2 or computer, you must enable and set up the Windows Remote Desktop Session Host.

For the most up-to-date Remote Desktop Session Host information for Windows[®] Server 2008 R2, visit the Microsoft[®] web site at:

[http://technet.microsoft.com/en-us/library/dd647502\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/dd647502(WS.10).aspx)



Important:

Before installing the Workflow Remote Desktop Session Host, be aware of its limitations and software and licensing requirements. You should also spend some time planning a sound network and security strategy, and user and machine naming conventions.

Workflow Remote Desktop Session Host Limitations

The Workflow Remote Desktop Session Host has a number of limitations and assumptions.

- Location-based functionality in Workflow does not work with Remote Desktop Services. That is, workflow tasks that are assigned to a location, key set permissions assigned to a specific computer, and any computer configurations, including layout and auto login/logout, will not work on a Remote Desktop Session Host.
- The Remote Desktop Session Host computer must have Windows[®] Server 2008 R2 or Windows[®] Server 2012 installed.
- Workflow does not support running a Remote Desktop Session Host on Windows[®] Vista.
- Although Windows supports machine names of up to 15 characters, Workflow node names are limited to eight characters.
- Depending on your settings, some keyboard shortcuts (such as Ctrl+Alt+Delete) may be disabled or remapped.
- User accounts must be created with logins after installing Workflow.

Scalability

The number of clients supported by a server varies according to the server's processing power and memory. If you use good optimization techniques, you should be able to run more clients with better performance per server.

The Microsoft[®] Performance Monitor can help you to determine the optimal number of Remote Desktop Session Host sessions your server can handle.

Hardware Requirements for Remote Desktop Session Host

The following information describes the recommended hardware requirements for a Remote Desktop Session Host environment.

- One 3Ghz CPU for every three clients connected to the Remote Desktop Session Host (*minimum*)

**Note:**

If the clients are expected to manage high levels of activity, decrease the number of clients per CPU to two.

- 500MB RAM per client (*minimum*)

Software Requirements for Remote Desktop Session Host

Workflow Remote Desktop Session Host requires Workflow and the Windows[®] Server 2008 R2 Standard or Enterprise Edition or the Windows[®] Server 2012 operating system.

**Note:**

Ge Digital highly recommends that you install the latest Service Pack and Windows Updates.

Licensing Requirements

GE Digital recommends that you install your Remote Desktop Session Host on a computer that is not a domain controller. Ensure that you back up the Remote Desktop Session Host licenses regularly to prevent data loss.

For the most up-to-date Remote Desktop Session Host license information on Windows[®] Server 2008 R2, visit Microsoft[®]'s web site:

[http://technet.microsoft.com/en-us/library/dd647502\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/dd647502(WS.10).aspx)

Workflow Remote Desktop Session Host Clients for Windows[®] Server 2008 r2

When a Windows[®] Server 2008 R2 client connects to the Remote Desktop Session Host with External Connector Licensing disabled, the license from the client is used. The first time that you connect to the server, that license is activated.

If you are running a non-Windows[®] Server 2008 R2 client computer, you need to purchase Windows Remote Desktop Services Client Access Licenses (RDSCALs) from Microsoft[®]. When you first set up the Workflow Remote Desktop Session Host, you have a 120-day temporary license to run a maximum of 10 client machines from the server. The first 10 Windows client machines that connect to the server will reserve the licenses.



Note:

Remote Desktop Services Client Access Licenses (RDSCALs) are counted per device and are not concurrent.



Important:

When you activate a RDSCAL, it is permanently associated with that machine. You cannot reboot the machine to clear your license use. If you reformat the machine, you must contact Microsoft® to obtain a replacement license.

Workflow Licensing

All hardware and software keys distributed to run Workflow contain software to enable licensing for Remote Desktop Session Host client computers.

For example, if you purchase a ten-client license, software embedded in the key allows ten concurrent client users and prevents an eleventh client access to Workflow. When a client disconnects from Remote Desktop Session Host, another client can then access it.



Note:

You still need enough RDSCALs to license all the devices that will connect to the server. Be aware that even if you have the appropriate number of Remote Desktop Session Host licenses enabled, performance issues may prevent you from connecting.

Enable Windows Server 2008 R2 Remote Desktop Services

Remote Desktop Services must be enabled on the Windows® Server 2008 R2 computer that will be used as the Remote Desktop Session Host.

About this task



Important:

These steps must be performed before you install Workflow.

For more information about Remote Desktop Session Host, refer to the Remote Desktop Services section of the Windows® Server 2008 R2 Help or visit Microsoft's web site at: [http://technet.microsoft.com/en-us/library/dd647502\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/dd647502(WS.10).aspx).

Procedure

1. On the Windows® Server 2008 R2 computer, click **Start > Administrative Tools > Server Manager**.

The Server Manager dialog box appears.

2. Click **Roles**, and then click **Add Roles**.

The **Select Server Roles** page of the **Add Roles Wizard** appears.

3. Select **Remote Desktop Services**, and then click **Next**.

The **Select Role Services** page appears.

4. Select the **Remote Desktop Session Host** and **Remote Desktop Licensing** check boxes.

You can select additional role services, but no other services are required.

**Important:**

If your application server is configured to use AD LDS (Active Directory Lightweight Directory Services), a warning message appears. Click **Install Remote Desktop Session Host anyway (not recommended)**.

5. Click **Next**.

The **Specify Authentication Method for Remote Desktop Session Host** page appears.

6. Select an option to indicate whether network level authentication is required, and then click **Next**.

The **Specify Licensing Mode** page appears.

7. Select an option to specify the Remote Desktop licensing mode that you want this Remote Desktop Session Host server to use, and then click **Next**.

The **Select User Groups Allowed Access to this RD Session Host Server** page appears.

8. Click **Add** to add the required users or user groups that can connect to this Remote Desktop Session Host server, and then click **Next**.

The **Configure Client Experience** page appears.

9. Select the functionality you want to provide to the users connecting to a remote desktop session, and then click **Next**.

The **Configure Discovery Scope for RD Licensing** page appears.

10. **Optional:** Select the **Configure a discovery scope for this license server** check box, and then configure it as follows:

- a. Select the appropriate option for the discovery scope.
- b. Click **Browse** to select the location for the RD Licensing database.

11. Click **Next**.

The **Confirm Installation Selections** page appears.



Note:

At any time during the configuration process, you can click **Previous** to go to a previous page to change your settings.

12. Click **Install**.

If instructed to do so, restart the computer for the changes to take effect.

Log into a Remote Desktop Session Host from Windows 7

Procedure

1. Click **Start > All Programs > Accessories > Remote Desktop Connection**.

The **Remote Desktop Connection** dialog box appears.

2. Enter or select a computer name, and then click **Connect**.

The **Log On to Windows** dialog box appears.

3. Enter a user name and password, and then click **OK**.

Related information

[Configure a Remote Desktop to Connect to a Remote Desktop Session Host \(on page 180\)](#)

Workflow with Windows Remote Desktop Session Host Installation and Configuration

After enabling Installing Remote Desktop Services and its licensing, you can install and configure Workflow on a Remote Desktop Session Host.

Prerequisites for Workflow on a Remote Desktop Session Host

Prior to installing any applications for use with Remote Desktop Session Host clients, refer to the Microsoft[®] Remote Desktop Session Host documentation regarding concept, configuration, and use of ROOTDRIVE.

Before you begin

Before you begin installing and configuring Workflow on a Remote Desktop Session Host, GE Digital strongly recommends that you complete the following Windows administrative tasks on the Workflow server machine.

About this task**Important:**

If you do not follow the recommended steps for the correct Windows Remote Desktop Session Host Configuration, your applications may not function correctly.

Procedure

1. From the `C:\Windows\Application Compatibility Script` folder, run **chkroot.cmd**.
The `RootDrv2.cmd` file is created.
2. In a text editor, open the `RootDrv2.cmd` file from the `C:\Windows\Application Compatibility Script` folder.
3. At the end of the file, on the `Set RootDrive=` line, add a drive letter.
For example: `Set RootDrive=W:`
4. Save the `RootDrv2.cmd` file.
5. Run the `RootDrv2.cmd` file.
6. Verify that your Remote Desktop Session Host is set up and functioning.

Install Workflow on a Remote Desktop Session Host

Follow the installation steps, as outlined in the Workflow Getting Started Guide, to install Workflow on a Remote Desktop Session Host.

Specify the program that starts when the user logs on to the Remote Desktop Session Host

You can create a secure environment that prevents operators from performing unauthorized actions, such as using the Ctrl+Alt+Delete key combination to shut down the Remote Desktop Session Host. You can configure this in Windows, with user properties and group policies.

About this task

Be aware that you can use Efficiency Analyzer security to define the rights each user has in Efficiency Analyzer after logging in. For example, you can add further restrictions in Efficiency Analyzer by setting permissions for the group the user is a member of.

Procedure

1. On the Remote Desktop Session Host machine, log in to Windows as an administrator.
2. Click **Start > Settings > Control Panel > Administrative Tools > Computer Management**.
The Computer Management dialog box appears.
3. In the System Tools folder, double-click **Local Users and Groups**.
4. Select the **Users** folder.

5. Right-click the user you want to configure, and then select **Properties**.
The **Properties** dialog box appears.
6. Click the **Environment** tab.
7. In the Starting Program area, select the **Start the following program at logon** check box.
8. In the **Program File Name** field, enter the path to the Workflow client program.

For example, if you installed Workflow to the default location, enter:

```
C:\Program Files\Proficy\Proficy Workflow\Proficy Client.exe
```

9. In the **Start in** field, enter the path of the Efficiency Analyzer product.

For example, if you installed Workflow to the default location, enter:

```
C:\Program Files\Proficy\Proficy Workflow
```

10. Click **OK**.
11. Repeat steps 5 through 10 for each user you want to configure.
12. Click **OK** to close the Computer Management dialog box.

Configure a Remote Desktop to Connect to a Remote Desktop Session Host

Windows[®] Server 2008 R2, Windows[®] 7, and Windows[®] Server 2012 provides the Remote Desktop Connection to configure and connect to a remote Remote Desktop Session Host.

Procedure

1. Click **Start > All Programs > Accessories > Remote Desktop Connection**.
The **Remote Desktop Connection** dialog box appears.
2. Click **Options**.
3. On the **General** tab, in the **Computer** field, enter or select a name from the drop-down list.
4. In the **User name** field, enter a user name and, if applicable, preface it with your domain name followed by a backslash.
5. Click the **Display** tab.
6. Leave the default remote desktop size but, if required, change the color setting.
7. Click the **Experience** tab.
8. Select the performance options that your client requires. GE digital recommends that you clear all check boxes except **Persistent bitmap caching**.
9. Select the **Reconnect if connection is dropped** check box.
10. Click **Connect** to connect to the Remote Desktop Session Host.

Troubleshoot Your Workflow Remote Desktop Session Host Environment

To successfully troubleshoot your Workflow Remote Desktop Session Host environment, you first need to isolate the source of your problem.

Troubleshoot Specific Remote Desktop Session Host Problems

After isolating your problem area, use the information in the following table to begin troubleshooting.

For this problem area...	Verify...
Performance	Should you run fewer clients? Server performance increases with the following: <ul style="list-style-type: none"> • Faster processors • More processors (dual, quad) • More memory • Reduced graphic refresh rate on clients
Connection	<ul style="list-style-type: none"> • Is your connection problem for all connections or user specific? • Can the Administrator start a session locally and remotely?
Environment	<ul style="list-style-type: none"> • Can you create a session on the server? • Is the problem the same for all users including Admin? • Can you create a new client session? • Can you create a client that opens Notepad?

For more information, suggestions, and tips for troubleshooting your Workflow Remote Desktop Session Host environment, refer to the Windows[®] Server 2008 R2 Help and the Microsoft[®] Remote Desktop Session Host online documentation.

Troubleshoot Known Remote Desktop Session Host Problems

The following table lists some specific troubleshooting information for known Remote Desktop Session Host problems.

Problem	Solution
Registry locks, behaves like "Duplicate Node Names" on the network.	You have started multiple sessions at the same time or multiple users have logged on using the same account. Multiple users cannot log on using the same account. Start one session at a time.

Problem	Solution
	If the same user is logged on multiple times, consider using the End Session setting. The End Session setting logs off the user upon disconnect. To use this setting, select the End Session check box located under Override User Settings in the RDP Properties of the Internet Information Services (IIS) Manager Configuration Console.
The text insertion cursor in a Workflow Text object does not appear.	The cursor blink rate in the control panel is set to the lowest setting for Remote Desktop Session Host. Set the blink rate to 1 above the slowest setting.

Frequently-Asked Questions About Using Workflow with Remote Desktop Session Host

The following list outlines some of the frequently-asked questions about using Workflow with Remote Desktop Services.

How can I determine the number of Remote Desktop Session Host connections that my Workflow license supports?

If you have an M4 key, run the Proficy License Viewer (Proficy.ManagementConsole.exe). Select the Workflow product icon to display Workflow information.

ActiveX Task Controls

Task Controls in GE HMIs

The Workflow Task List can be integrated into other GE HMI applications.

Workflow task controls can be integrated into existing HMI applications, such as HMI/SCADA CIMPLICITY and HMI/SCADA iFIX. Workflow tasks are displayed in your application so you can operate them from within these applications. You use the ActiveX Task List and Task Indicator controls to interact with and view workflow tasks. The ActiveX Task List provides users with a list of tasks for a workflow. The Task Indicator provides a user with relevant information regarding the status of the ActiveX Task List.

Task List Configuration

The configuration tool allows you to modify server information and its authentication credentials.

Before configuring the ActiveX Task List, note the following information:

- Each installation of the ActiveX Task List can connect to only one Workflow server at a time.
- At a single workstation, only one Task Indicator object and one ActiveX Task List object can be connected at a time per HMI application instance.
- Workflow task controls and GE HMIs are related, but work independently from each other. Therefore, user names and logins are separate.
- Like the Workflow client, the NET. TCP Port Sharing service must be enabled to use the Task List in a host application.
- To change the server name, use the Configure Client tool. For more information, see **Modify the server configuration for a remote client**. If you install the ActiveX Task List on a different machine than the application server, you must access the Configure Client tool from the following location:
`../Program Files/Proficy/Proficy Task List/Program.`

Install the ActiveX Task List

Before you begin



Important:

Product Options Installation Information. When using the ActiveX Task List, product option updates made on the server are not automatically deployed to applications using the Task List. The `ProficyUpdater.exe` application must be run manually, with administrative privileges, in order to deploy production option updates on each client machine using ActiveX Task List.



Note:

At a single workstation, only one Task Indicator object and one ActiveX Task List object can be connected at a time per HMI application instance.

Procedure

1. From the application splash screen, click **Install ActiveX Task List**.



Note:

If the splash screen does not appear, double-click `SetupTaskList.exe` from the root directory of the installation folders.

The **License Agreement** page appears.

2. Review the license agreement, and then click **I Agree**.

The **Installation Folder** page appears.

3. Accept the default destination folder or browse for a new location, and then click **Next**.
The **Proficy Workflow Server Configuration** page appears.
4. In the **Server Name** field, enter the name of the server that you want to install the ActiveX Task List on.
The **Security Certificates** page appears.
5. The security certificates are automatically downloaded from the server you are connecting to.
6. Click **Next**.
The **Configure Task List Authentication** page appears.
7. From the **Authentication Type** drop-down menu, select one of three login options, and then click **Next**.
If you want the user name and password:
 - entered manually, select **Manual Authentication**, or
 - exposed in plain text, select **Control Property Authentication**, or
 - saved automatically after initial setup, select **Password Encryption Authentication**.



Note:

You must first configure users using the **Configure Client** tool.

The **Installation Confirmation** page appears.



Note:

At any time during the configuration process, you can click **Back** to return to a previous page to change your settings.

8. Click **Install**.
The Installation page appears, displaying the status of each installation step.
9. When the installation is complete, click **Exit**.

Modify the ActiveX Task List installation configurations

Before you begin



Note:

To change the server name, use the Configure Client tool. For more information, see **Modify the server configuration for a remote client**. If you install the ActiveX Task List on a different machine than the application server, you must access the Configure Client tool from the following location: `../Program Files/Proficy/Proficy Task List/Program`.

Procedure

1. From the **Start** menu, select the **Task List Configuration**.

The **Proficiency Task List Configuration** page appears.

2. To add or modify a server, in the **Server Name** section, enter the fully qualified name of a server.
3. To modify the login authentication, in the **Server Authentication** section:

Select...	To...
Manual Login	prompt the user to enter a user name and password at each login.
Control Property	allow the user to view the user name and password in the host's object properties.
Password Encryption	log the user in automatically.

4. If you selected **Password Encryption** in the **Server Authentication** section, add user credentials in the **User Configuration** section:
 - a. In the **Users** panel, click **Add**.
 - b. In the **Details** panel, in the **User Name** field, enter an existing Workflow user name.
 - c. In the **Password** field, enter your user account password.
 - d. In the **Confirm Password** field, enter your password again.
5. Click **Save**.

Workflow Task Client

Overview

This topic provides an overview on how to install and start working with the workflow task client in the Operations Hub application.

- [Install the workflow task client \(on page 186\)](#) from the ISO file.
- [Configure the task client \(on page 194\)](#) widget in Operations Hub.
- [Access the task client \(on page 197\)](#) features in Operations Hub.

Related reference

[Troubleshooting Task Client Issues \(on page 207\)](#)

Related information

[Install the Task Client \(on page 186\)](#)

[Uninstall the Task Client \(on page 202\)](#)

[Generate an Equipment Model for Web HMI \(on page 422\)](#)

[Import Equipment Model \(on page 196\)](#)

[Create an User Account in Operations Hub \(on page 196\)](#)

[Add Users and Personnel Classes \(on page 340\)](#)

Install the Task Client

This information guides you through the process of installing the task client on the Operations Hub server.

Before you begin

Install Operations Hub on the machine you want to run the task client installer. For the latest compatibility information on Operations Hub, refer to 'Compatibility and Upgrading with GE Products' section in the *Getting Started* Guide.



Note:

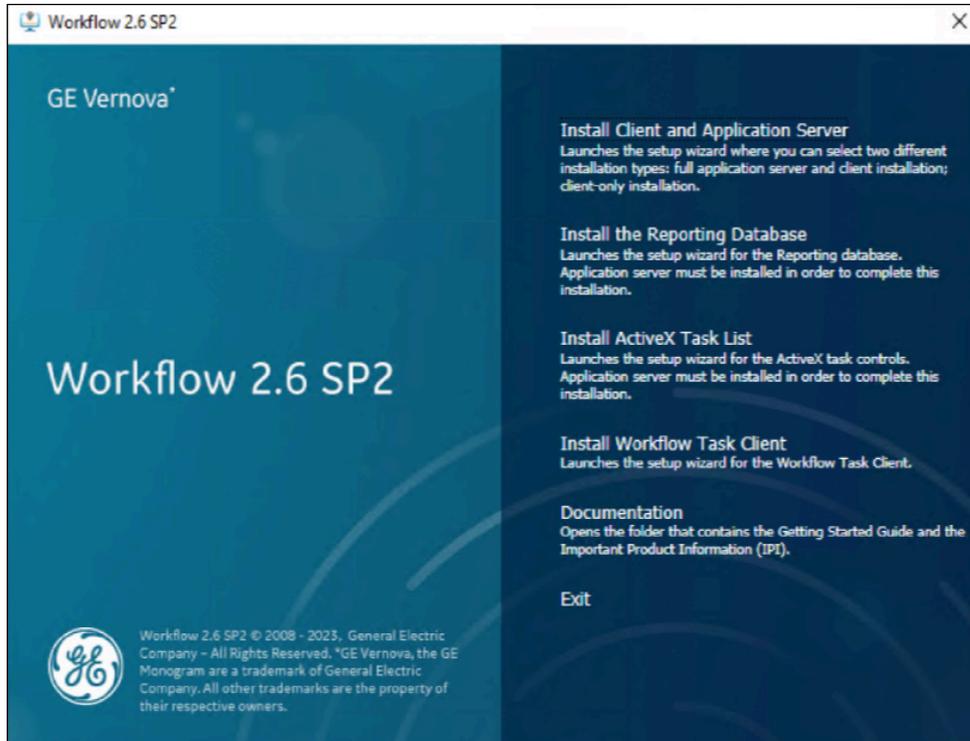
For instructions, refer to the Operations Hub UAA Installation Guide provided along with the Operations Hub installation package.

About this task

The task client installer is shipped as a part of the Workflow 2.6 SP2 installation package.

Procedure

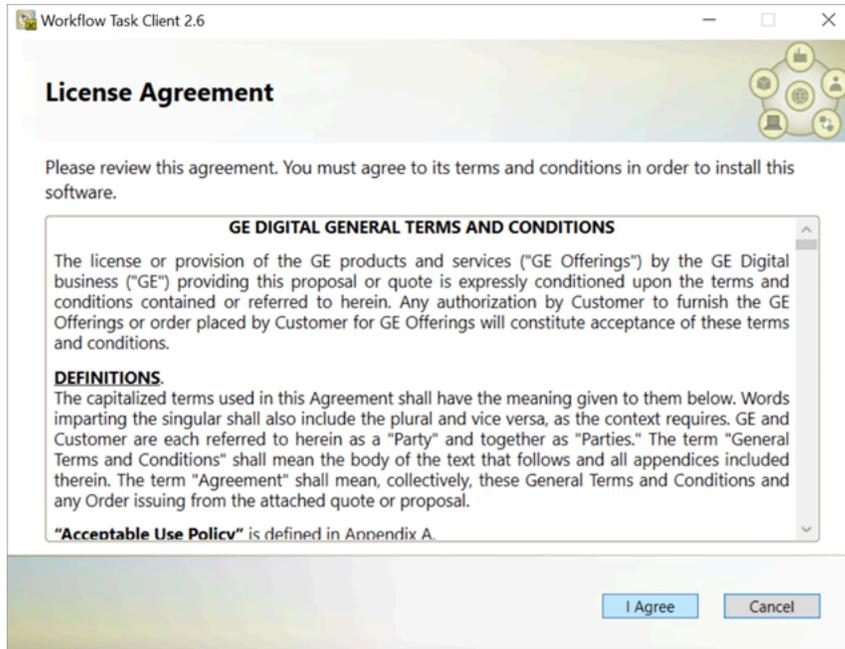
1. Copy the Workflow 2.6 SP2 ISO file to the machine where you want to install the task client, and run the file.
2. From the application splash screen, click **Install Workflow Task Client**.

**Note:**

If the splash screen does not appear, run `InstallFrontEnd.exe` on the root directory of the installation folder.

The **License Agreement** page appears.

3. Review the license agreement, and then click **I Agree**.



The **Workflow Server** page appears.

4. Enter the details as specified in the following table:

Field	Description
Server Name	Enter the host name or IP address of the Workflow server.
REST Port	Enter the Workflow port number that allows the REST API calls. The task client widget in Operations Hub uses this port to request data from the Workflow application. 8447 is the default Workflow port number.

Workflow Server

Please specify the Workflow server details for this installation.

Server Details

Server Name: wftrunkl

REST Port: 8447

Note:
The Workflow Task Client plugin import may fail if the above details are not correct.

<Back Next> Cancel

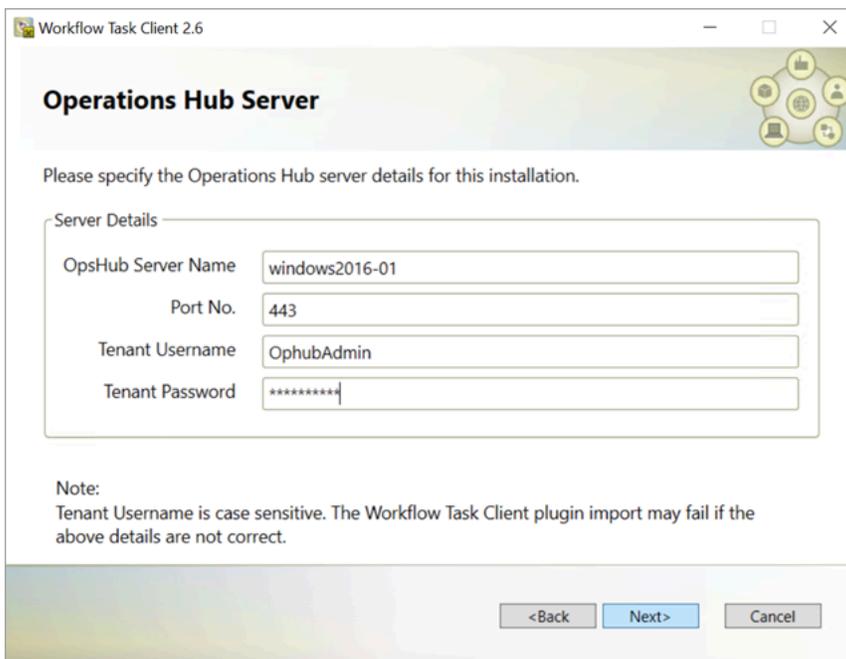
5. Click **Next**.

The **Operations Hub Server** page appears.

6. Enter the details as specified in the following table:

Field	Description
OpsHub Server Name	<p>Enter the Operations Hub server name (FQDN or host name) as per the configuration of UAA server.</p> <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: Ensure that the auto populated OpsHub server name is correct.</p> </div>
Port No.	<p>Enter the Operations Hub UAA server port number. 443 is the default UAA port number.</p>
Tenant Username	<p>Enter the username of the administrator who has read-write permissions to access the Oper-</p>

Field	Description
	<p>ations Hub database. <code>OphubAdmin</code> is the default tenant username.</p> <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: The tenant username is case-sensitive.</p> </div>
Tenant Password	Enter the password for the administrative user you entered in the Tenant Username box.



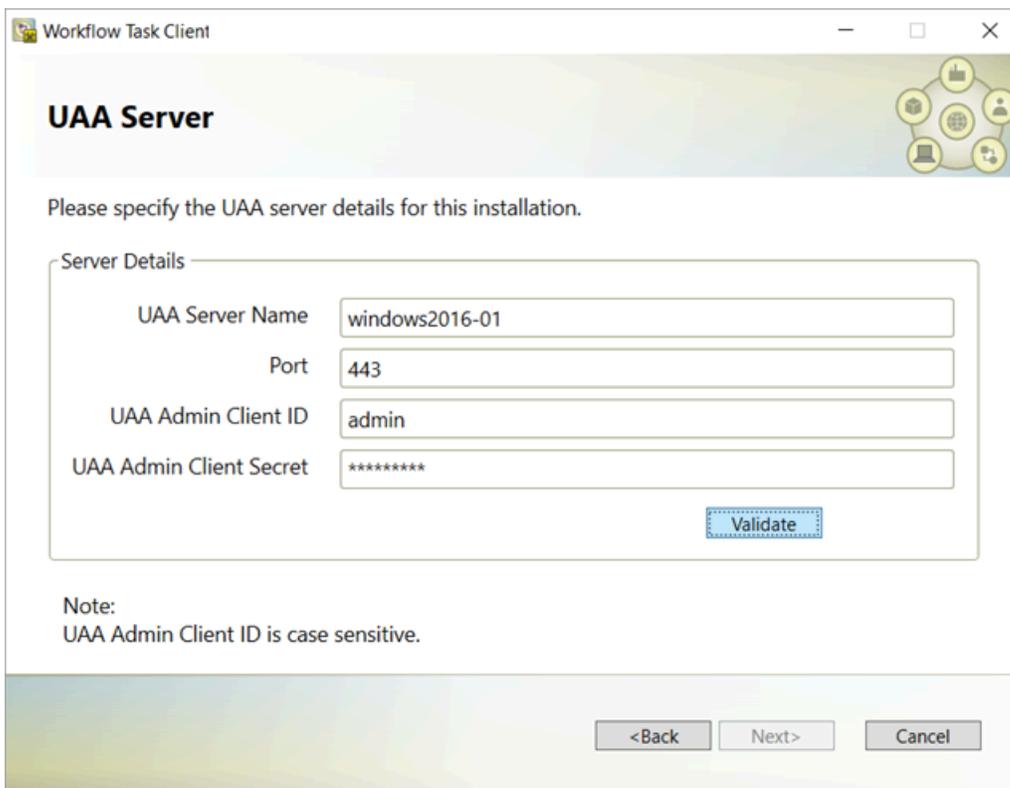
7. Click **Next**.

The **UAA Server** page appears.

8. Enter the UAA Server details for authenticating the users to access the Workflow task client as specified in the following table.

Field	Description
UAA Server Name	Enter the Operations Hub UAA server name (FQDN or host name) as per the configuration of UAA server.

Field	Description
	 Note: Ensure that the auto populated UAA server name is correct.
Port	Enter the Operations Hub UAA server port number.
UAA Admin Client ID	Enter the client ID of the Operations Hub UAA administrator.  Note: The UAA admin client ID is case-sensitive.
UAA Admin Client Secret	Enter the client secret of the Operations Hub UAA administrator.



The screenshot shows a window titled "Workflow Task Client" with a sub-header "UAA Server". Below the header, it says "Please specify the UAA server details for this installation." There is a "Server Details" section with four input fields: "UAA Server Name" (windows2016-01), "Port" (443), "UAA Admin Client ID" (admin), and "UAA Admin Client Secret" (masked with asterisks). A "Validate" button is located below the secret field. At the bottom, there is a "Note: UAA Admin Client ID is case sensitive." and three navigation buttons: "<Back", "Next>", and "Cancel".

9. Click **Validate** to verify the UAA server credentials.

A message appears to inform whether the user account authentication is successful. If **PASS**, then you can proceed to install workflow task client. If **FAIL**, then you must enter the correct credentials.

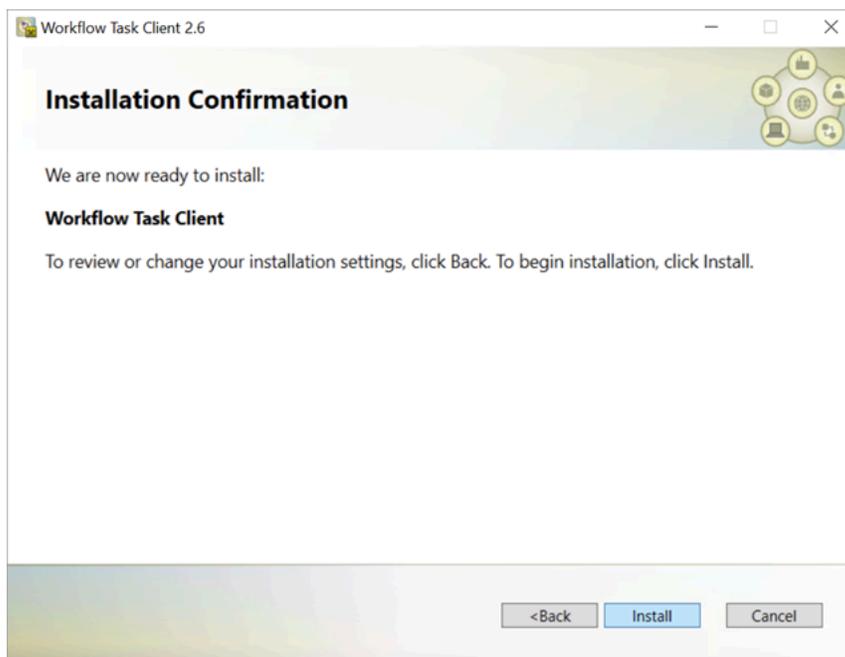
10. Click **Next**.

The **Installation Confirmation** page appears.



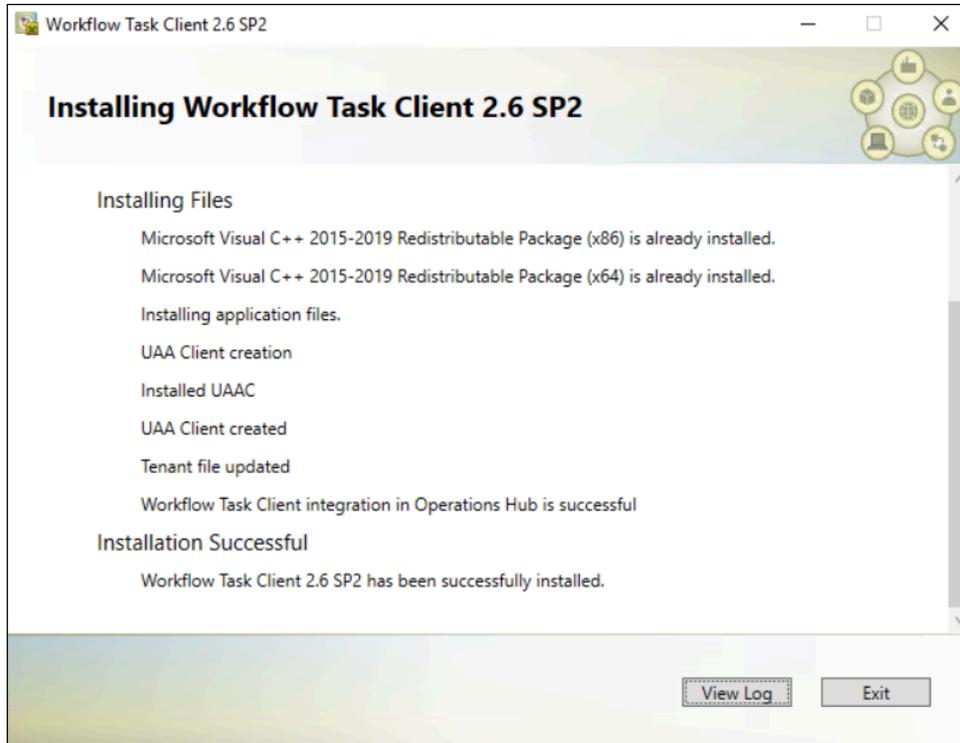
Note:

At any time during the configuration process, you can click **Back** to return to a previous page to change your settings.



11. Click **Install**.

The Installing Workflow Task Client 2.6 SP2 page appears, updating the status of the installation.



12. When the installation is complete, click **Exit**.

The task client is added to the Operations Hub application.

Results

The task client is added with the following information in Operations Hub:

- The Workflow Tasklist application is available under **APPS**.
- The entity `WF_User_Credentials` is created under **ENTITIES** to save user logins.
- The queries `DeleteWorkflowUser`, `getWorkflowUser`, and `InsertWorkflowUser` are created under **QUERIES**.
- `TASK LIST` and `TASK COUNT` are added as system widgets in Operations Hub.

What to do next

[Configure the task client \(on page 194\)](#) widget in Operations Hub.

Related reference

[Troubleshooting Task Client Issues \(on page 207\)](#)

Related information

[Uninstall the Task Client \(on page 202\)](#)

Configure the Task Client

This topic describes how to enable the task client widget to operate in the Operations Hub web server environment.

Before you begin

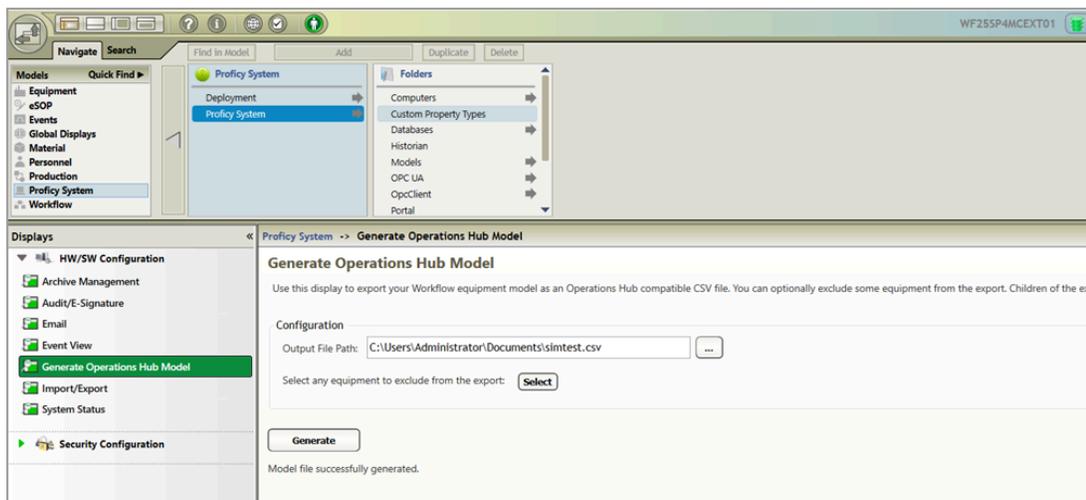
Install the Workflow and the Operations Hub applications on different servers.

About this task

To configure the task client widget, equipment models from the Workflow application are exported to a .csv file, and imported into the Operations Hub application.

Procedure

1. In the Workflow server, do the following:
 - a. Select an equipment model you want to export, and then select **Generate Operations Hub Model**.
 - b. Provide a location to save the exported file, and select **Generate**.



The equipment model data is exported to a .csv file format, and saved in the specified location.

2. In the Operations Hub server, do the following:
 - a. [Import the .csv file created in step 1 \(on page 196\)](#) to Operations Hub.
 - b. In the main navigation menu, select **APPS**, and then select Workflow Tasklist application. The Pages workspace appears.
 - c. Select **Tasklist View**.

The pages associated to the task list appear in a container. `TASK LIST`, and `TASK COUNT` are system widgets that cannot be deleted.

d. Select the **Task List** page in the container.

The **GETASKLIST PROPERTIES** tab settings appears on the right pane.

e. Provide **DATA** settings as specified in the table below:



Note:

Scroll down in the settings section to find the data settings after the general and display settings.

Parameter	Selection	Description
WorkflowServer	Manual	Enter the URL address of the Workflow server to connect.
RefreshRate	Manual	Enter the time in seconds at which rate the task client will refresh to get the latest data from the Workflow server (for example, 5).
Height	Manual	Enter the preferred height of the task client widget (for example, 600).

f. Select **Save App**.

The application settings are saved.

What to do next

- Create identical user accounts in Workflow and Operations Hub.
- [Log in to the Operations Hub web client \(on page 197\)](#) to connect and work with the task client widget.

Related information

[Generate an Equipment Model for Web HMI \(on page 422\)](#)

[Import Equipment Model \(on page 196\)](#)

[Create an User Account in Operations Hub \(on page 196\)](#)

Import Equipment Model

By importing the Workflow equipment models, you can manage the tasks assigned to you from within the Operations Hub application.

Before you begin

Ensure that the Workflow equipment models are exported to a Web HMI model .csv file.

About this task

An equipment model is configured in the Workflow application. For more information, refer to the Equipment Model topic in the *Resource Information and Configuration* section of the complete Workflow user guide.

Procedure

1. Access Operations Hub.
2. In the main navigation menu, select **ADMIN**.
3. In the Admin workspace, select **Import/Export**.

The **Model Import/Export** page appears.



Tip:

Select  to hide the navigation pane that overlaps the page.

4. To import, browse and select the exported .csv file from Workflow, and select **Import**.

Results

The imported information is visible under Objects and Object Types in the Admin workspace.

Related information

[Generate an Equipment Model for Web HMI \(on page 422\)](#)

Create an User Account in Operations Hub

This topic describes how to create an user account in Operations Hub.

Before you begin

Only a tenant administrator can create and manage developers.

Procedure

1. In the main navigation menu, select **MANAGE**, and then select **Developers** or **App Users**.
2. Select **Add new user**.

The **New Account** window appears.

3. Enter values in the available fields as described in the following table.

Field	Description
Username	Enter the user name that the user will use to log in to Operations Hub. The value must be unique.
E-mail	Enter the email ID of the user. The value must be unique.
First Name	Enter the first name of the user.
Last Name	Enter the last name of the user.
Password	Enter a password that the user will use to log in to Operations Hub. The password must meet the following criteria: <ul style="list-style-type: none"> ◦ Must contain between 8 and 15 characters ◦ Must include at least one number ◦ Must include at least one uppercase or lowercase letter
Repeat Password	Enter the password that you have entered in the Password field.
Only GE Groups	Select this check box if you only want to view groups associated with GE products in the Groups list box.
Groups	Select <code>iqp.user</code> group to assign to this user.
Apps	Select <code>Workflow Tasklist</code> to allow the user to access this application.

4. Select **Create**.

The user is created. If you have created a developer, an application user is also created.

Access the Task Client

Use the Task Client widget to display task lists from the Workflow application in Operations Hub.

Before you begin

Create identical user accounts for the Operations Hub web server and the Workflow server. You must be able to log in to both the servers using the same username and password combination.

About this task

The Workflow Task List is integrated with Operations Hub. You do not have to log in to the Workflow application to manage your tasks. You can log in to the Operations Hub application, and connect to Workflow with the help of the task client widget to manage tasks.

Procedure

1. Log in to the Operations Hub web client.

The login page to connect to the Workflow server appears.

Username: admin

Password:

Server Address: https://wftrunkl:8447/

Log On

2. Enter the details as specified in the following table:

Field	Description
Username	The account username that has permission to access the Workflow application.
Password	The password for the username you entered in the Username box.
Server Address	The URL address to connect to the Workflow server. This URL is populated based on the properties provided for the Task Client widget in Operations Hub.

3. Select **Log On**.

The task count icon indicates the status of your workflow connection.

Status	Description
	Indicates that the Workflow server is disconnected due to either of these reasons:

Status	Description
	<ul style="list-style-type: none"> ◦ You are not logged in to the server. ◦ Internet connection is lost. ◦ Workflow server is down. <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: All tasks in Operations Hub remain disabled until a connection is established. </div>
	<p>Indicates that the Workflow server is connected.</p> <p>Once connected, the task count shows the number of workflow tasks. If the count is 0, it means there are no workflows.</p>

The user login credentials are encrypted and stored under **ENTITIES** in Operations Hub. You will not be prompted for login details again for the saved user accounts.



Note:

Whenever the Workflow server is down, or if there is no internet, the lost connection status is indicated in red. All tasks in Operations Hub remain disabled until a connection is established.

4. Select  to show the workflow equipment context navigation.

For more information, refer to the Equipment Context topic in the *Operator Task List in the Workflow Client* section of the complete Workflow user guide.

5. Select  or  to navigate and select any of these equipment contexts: Big Enterprise, Big Unit, or Big Area.

The equipment context set up shown in Operations Hub is created in the Workflow application. The list of tasks for the selected equipment context appears. Based on the list of tasks, the task count number  also gets updated.

**Note:**

Logged in users in Operations Hub can view only the tasks that are assigned to them in the Workflow application.

6. Use these options to work with the tasks:

Icon	Description
	Select to filter tasks or task steps by task name, priority, personnel assignment, step state, and expiry values. For more information, refer to the Task List Filtering topic in the <i>Operator Task List in the Workflow Client</i> section of the complete Workflow user guide.
	Select the Tasks for Equipment toggle to show or hide the tasks associated to the equipment location only.
	Select to start and run scheduled tasks. For more information, refer to the Start Task topic in the <i>Operator Task List in the Workflow Client</i> section of the complete Workflow user guide.
	Select to manually start a specific task step.
	Select to view and save a copy of the linked documents. For more information, refer to the Document(s) topic in the <i>Operator Task List in the Proficy Client</i> section of the complete Workflow user guide.
	Select to view the instructions to complete a task or task step. This option is available only when there are work instructions defined for the task or task steps.
	Select to access the following menus, and perform task related actions:

Icon	Description
	<ul style="list-style-type: none">◦ Set Priority: You can set a priority number for the task to run.◦ Jump to Task Step: You can skip some steps and jump to a specific task step in a scheduled flow.◦ Enter Expiry Comment: You can enter a reason for the task delay. The option to add a comment is available only after the task or task step has expired.◦ Cancel Task: You can cancel running a specific task or task step.◦ Reassign Personnel: You can reassign specific task steps to a different person or equipment location.◦ Acquire: You can acquire a specific task step if it is available.◦ Release: You can only release those steps that you have acquired. When you release a task step, it is available for other operators to acquire. <p>For more information, refer to the complete Workflow user guide.</p>
	<p>Select to access the forms attached to a task step, and update them. For more information, refer to the <i>Forms and User Displays Authoring Guide</i> section of the complete Workflow user guide.</p> <div data-bbox="862 1522 1424 1724" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;"><p> Note: You can load only HTML forms in the Operations Hub Task Client. Windows presentation framework (WPF), Sil-</p></div>

Icon	Description
	 verlight, and .NET rich client forms are not supported.

**Note:**

For added security, electronic signatures are configured in Workflow for specific tasks, task steps, or forms. In such cases, a dialog box appears requesting you to sign in for verification before accomplishing any task related action.

Related information

[Create an User Account in Operations Hub \(on page 196\)](#)

[Add Users and Personnel Classes \(on page 340\)](#)

Uninstall the Task Client

This topic describes the steps to uninstall the task client plugin.

Procedure

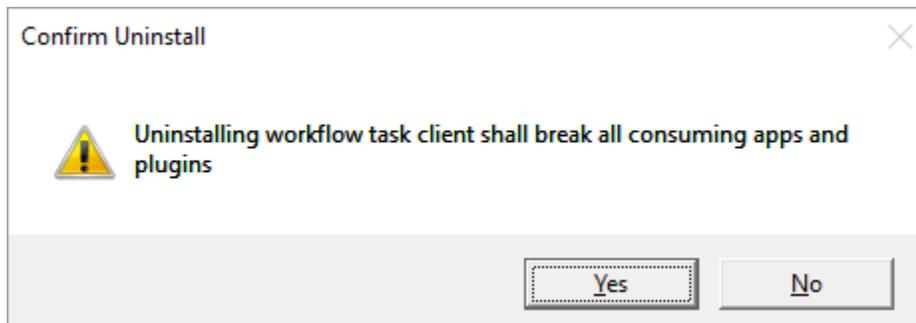
1. From the **Control Panel**, go to **Programs and Features**.

The **Uninstall or change a program** dialog box appears, displaying all the programs installed on the computer.

2. Select **Workflow Task Client**, and then **Uninstall**.
3. In the confirmation message box, click **Yes**.

The uninstallation process verifies once again to confirm the task client removal as it may lead to breaking any applications associated to it.

4. Click **Yes** to proceed with uninstallation.



A message appears when the uninstallation is complete.

- Click **OK** to close the message dialog box.

Uninstalling the task client removes the plugin information from Operations Hub **APPS, ENTITIES,** and **QUERIES**.

Results

Task Client is uninstalled.

Upgrade the Task Client

Before you begin

Install Operations Hub on the machine you want to run the task client installer. For the latest compatibility information on Operations Hub, refer to 'Compatibility and Upgrading with GE Products' section in the *Getting Started Guide*.



Note:

For instructions, refer to the Operations Hub UAA Installation Guide provided along with the Operations Hub installation package.

About this task

Use this procedure to upgrade Workflow Task Client from 2.6 SP1 to 2.6 SP2.

Procedure

- Copy the Workflow 2.6 SP2 ISO file to the machine where you want to install the task client, and run the file.
- From the application splash screen, click **Install Workflow Task Client**.



Note:

If the splash screen does not appear, run `InstallFrontEnd.exe` on the root directory of the installation folder.

The **License Agreement** page appears.

- Review the license agreement, and then click **I Agree**.

The **Workflow Server** page appears.

- View the following details:

Field	Description
Server Name	Displays the host name or IP address of the Workflow server.

Field	Description
	 Note: Ensure that the auto populated Workflow server name is correct.
REST Port	<p>Displays the Workflow port number that allows the REST API calls. The task client widget in Operations Hub uses this port to request data from the Workflow application. 8447 is the default Workflow port number.</p>  Note: Ensure that the auto populated REST port number is correct.

5. Click **Next**.

The **Operations Hub Server** page appears.

6. Enter the following details:

Field	Description
OpsHub Server Name	<p>Displays the Operations Hub server name (FQDN or host name) as per the configuration of UAA server.</p>  Note: Ensure that the auto populated OpsHub server name is correct.
Port No.	Displays the Operations Hub UAA server port number. 443 is the default UAA port number.
Tenant Username	Enter the username of the administrator who has read-write permissions to access the Operations Hub database. <code>OphubAdmin</code> is the default tenant username.

Field	Description
	 Note: The tenant username is case-sensitive.
Tenant Password	Enter the password for the administrative user you entered in the Tenant Username box.

7. Click **Next**.

The **UAA Server** page appears.

8. Enter the following details for authenticating the users to access the Workflow task client:

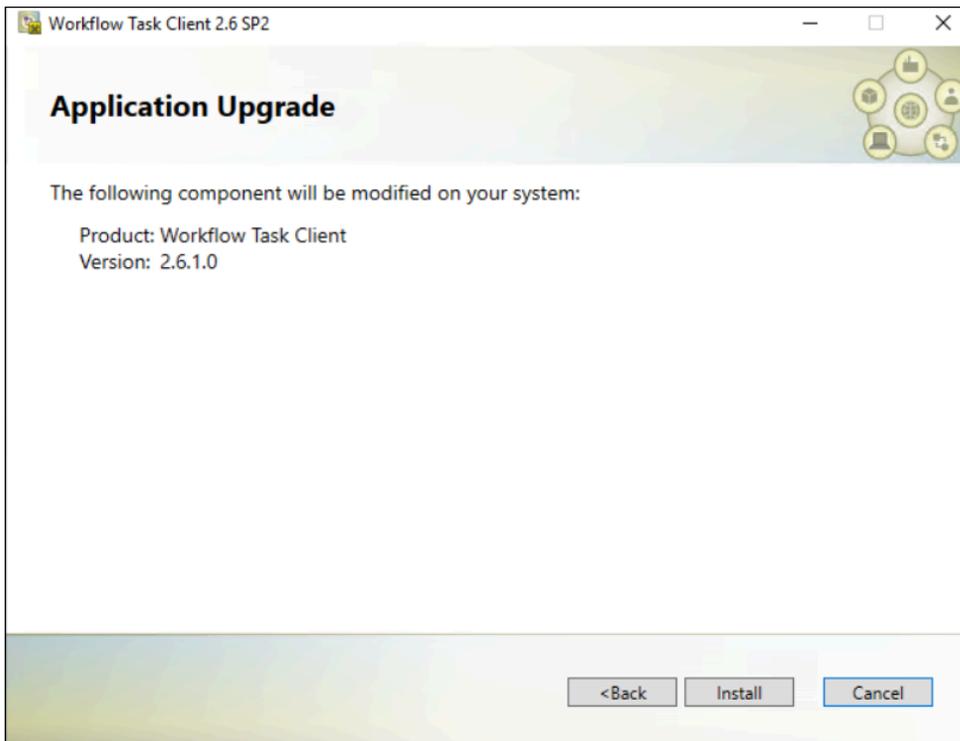
Field	Description
UAA Server Name	Displays the Operations Hub UAA server name (FQDN or host name) as per the configuration of UAA server.  Note: Ensure that the auto populated UAA server name is correct.
Port	Displays the Operations Hub UAA server port number.
UAA Admin Client ID	Enter the client ID of the Operations Hub UAA administrator.  Note: The UAA admin client ID is case-sensitive.
UAA Admin Client Secret	Enter the client secret of the Operations Hub UAA administrator.

9. Click **Validate** to verify the UAA server credentials.

A message appears to inform whether the user account authentication is successful. If **PASS**, then you can proceed to install workflow task client. If **FAIL**, then you must enter the correct credentials.

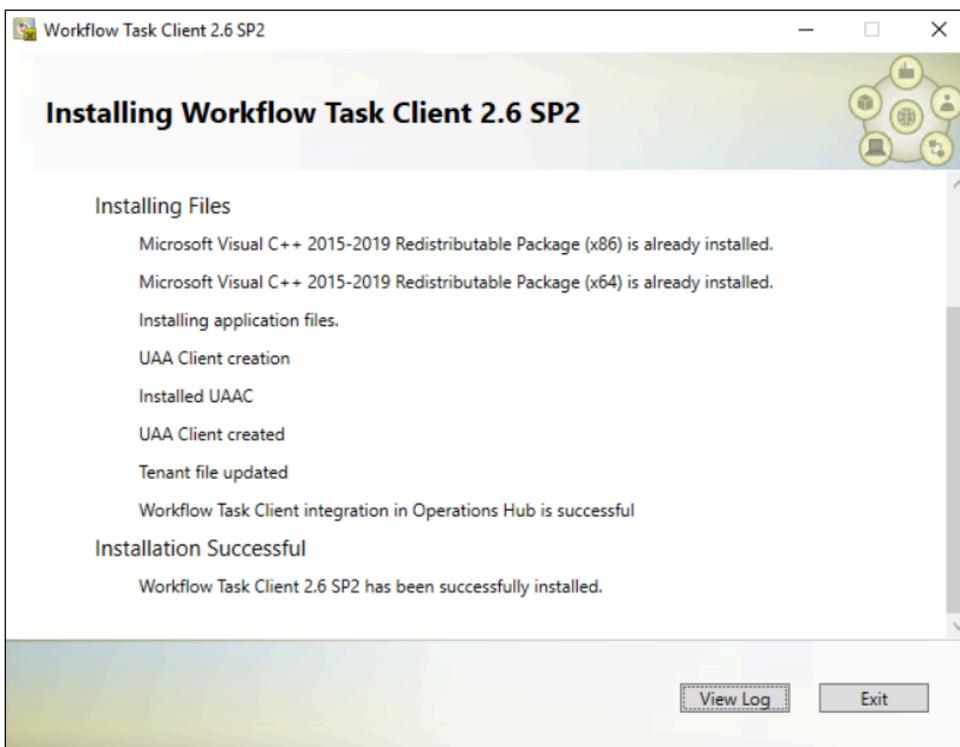
10. Click **Next**.

The **Application Upgrade** page appears.



11. Click **Install**.

The Installing Workflow Task Client 2.6 SP2 page appears, updating the status of the installation.



12. When the installation is complete, click **Exit**.

The task client is added to the Operations Hub application.

Results

The task client is added with the following information in Operations Hub:

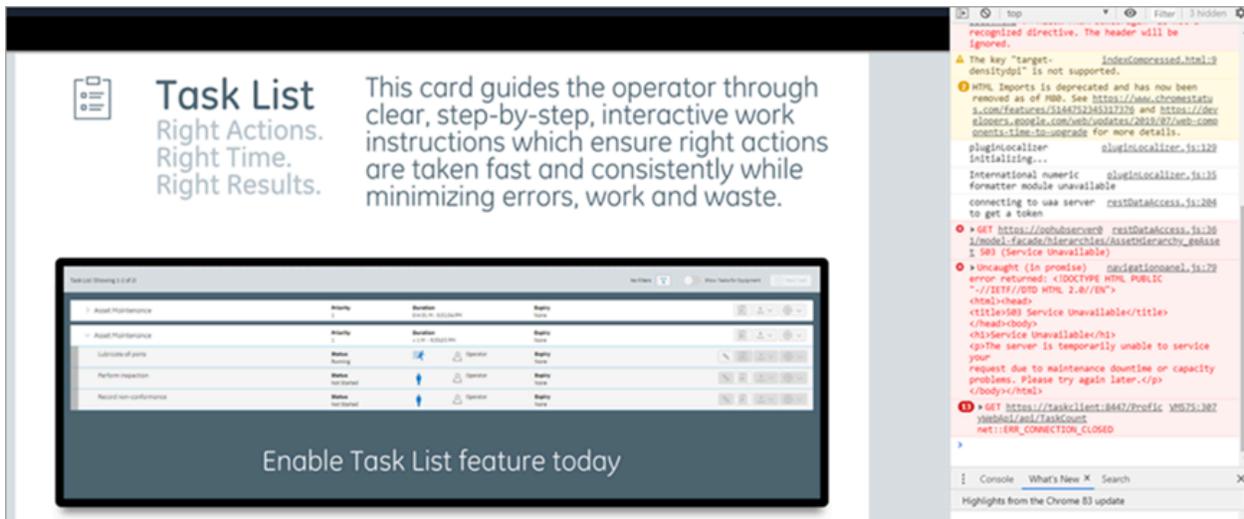
- The Workflow Tasklist application is available under **APPS**.
- The entity `WF_User_Credentials` is created under **ENTITIES** to save user logins.
- The queries `DeleteWorkflowUser`, `getWorkflowUser`, and `InsertWorkflowUser` are created under **QUERIES**.
- `TASK_LIST` and `TASK_COUNT` are added as system widgets in Operations Hub.

Troubleshooting Task Client Issues

This topic describes how to troubleshoot issues that you encounter when working with task client in Operations Hub.

Connection Errors

If this screen appears, either the internet connection is lost, or the Workflow server cannot be reached. On inspection, the `ERR_CONNECTION_CLOSED` implies that the connection to the Workflow is not established.



To troubleshoot the Workflow server connection errors, check for the following:

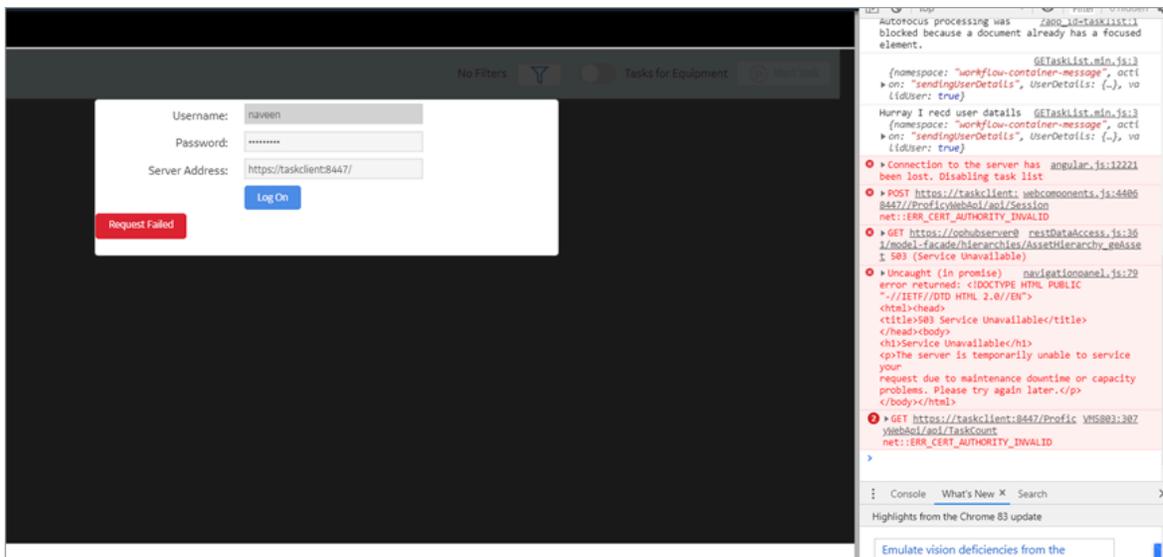
- Add the IP address of the Workflow server to your Windows hosts file: `c:\Windows\System32\Drivers\etc\hosts`.
- Ensure the Workflow server URL entered under **GETASKLIST PROPERTIES** in the Operations Hub administrative environment includes a trailing forward slash. For example, `https://taskclient:8447/`
- Verify the Workflow server URL saved for user login credentials under **ENTITIES** in Operations Hub also includes the trailing forward slash.

Security Certificate Errors

If using certificates that are signed by a Trusted Certificate Authority, you may not encounter this error.

If this screen appears, it is possible that you have a self-signed SSL Certificate. To troubleshoot the certificate issue, perform the following steps:

- In the browser inspect element, right-click the `ERR_CERT_AUTHORITY_INVALID` error link, and then click **Open in new tab**.



- In the new tab, click **Advanced**.
- Click the **Proceed to task client (unsafe)** link.
- Return to the task client application tab, and refresh the browser.

References to WebHMI

If using an older version of Workflow, you may come across these WebHMI references when you access the task client plugin properties. The references appear only when you add new instances of the task client plugin to application pages in Operations Hub.

- WebHMIUrl
- WebHMIUsername
- WebHMIPassword

To resolve the WebHMI references issue, upgrade to the Workflow 2.6 SP1 SIM 4 version.

Server Communication Error

If you are on Operations Hub 1.7 and directly apply task client 2.6 SP1 SIM 2 version, it leads to an error in communicating with the server. Operations Hub 1.7 communicates using `app_id`, whereas the task client looks for `app_name` to communicate with the server. This issue does not occur if you are on Operations Hub 2.0, which also communicates via `app_name` in its URLs.

To resolve the communication issue with Operations Hub 1.7, upgrade to the Workflow 2.6 SP1 SIM 4 version.

Operations Hub Interoperability



As a workaround to address VC++ (x64) failure, follow these steps to install Task Client on Operations Hub 2.1.

1. Uninstall the VC++ 2015-2019 Redistributable (x64) version installed with Operations Hub 2.1.
2. Open the Workflow install media, navigate to the `\Installers\Resources\Microsoft\2015_2019` folder, and run the `vc_redist.x64.exe`.
3. Open the Services console dialog and restart these services:
 - GE Operations IQP PostgreSQL Database
 - GE Operations UAA PostgreSQL Database
4. [Install the Task Client \(on page 186\)](#).
5. Install the VC++ 2015-2019 Redistributable (x64) version that got installed with Operations Hub 2.1.

Resolving Proficy Authentication Issue During Workflow Task Client Install

For Operations Hub 2022 release, Proficy Authentication server validation fails while installing Workflow Task Client. As a workaround to fix this issue, perform the following steps to successfully install Workflow Task Client with Operations Hub 2022.

1. Install Operations Hub 2022.
2. Modify the registry setting on the Operations Hub server to allow for TLS 1.2.
 - a. Access the following registry paths and modify the `DWORD` value name `SchUseStrongCrypto` with the value data as `1`.
 - `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\.NETFramework\v4.0.30319`
 - `HKEY_LOCAL_MACHINE\SOFTWARE\WOW6432Node\Microsoft\.NETFramework\v4.0.30319`
 - b. If the registry value does not exist, then create a new one with the same value name and value data.
3. [Install the Workflow Task Client \(on page 186\)](#) plugin from the Workflow installer on your Operations Hub Server host.



Note:

For more information, refer to [Microsoft suggested actions](#).

Installation Security

Installation Security

Security is the process of maintaining the confidentiality, integrity, and availability of a system.

- *Confidentiality*: Ensure only the people you want to see information can see it.
- *Integrity*: ensure the data is what it is supposed to be.
- *Availability*: Ensure the system or data is available for use.

GE recognizes the importance of building and deploying software with these concepts in mind and encourages customers to take appropriate care in securing their GE products and solutions.

More information on security, including GE security advisories and security patch notifications, can be found online.

Please visit our website at https://digitalsupport.ge.com/en_US/Alert/GE-Security-Advisories.

Defense in Depth

Defense in depth is the concept of using multiple layers of security to raise the cost and complexity of a successful attack.

To carry out a successful attack on a system, an attacker would need to find not just a single exploitable vulnerability, but would need to exploit vulnerabilities in each layer of defense that protects an asset.

For example, if a system is on a network protected only by a firewall, the attacker needs to circumvent only the firewall to gain unauthorized access. However, if there is an additional layer of defense such as a user name/password authentication requirement, the attacker needs to find a way to circumvent both the firewall and the user name/password authentication.

Anti-Virus Software

GE encourages customers to use third-party anti-virus (AV) software of their choice and to keep it current with the latest updates.

While GE does not specifically certify any particular AV supplier's software, we do test our products with GE's corporate standard (currently McAfee) installed and running on all test and system lab machines. In the event there is a Workflow product defect discovered while running any AV software, GE will make all reasonable efforts to provide a solution. However, if the issue is found to be based on specific behavior of the AV software, the customer might be advised to work with the AV software vendor and/or switch to another AV software vendor to get resolution to their issue.

Data Execution Prevention

GE products function with Microsoft Windows Data Execution Prevention (DEP) enabled, and GE recommends that customers enable this feature as an added protection against the exploitation of application security vulnerabilities such as buffer overflows.

In the event there is a Workflow product defect discovered while running DEP, GE will make all reasonable efforts to provide a solution.

Software Patching

Installing software patches is an integral part of security.

Patching GE Software

GE recommends that customers keep their software up-to-date by applying the latest Software Improvement Module (SIM) to their deployed GE products. SIMs add new functionality, fix bugs, and address security vulnerabilities. Security advisories and security-related SIMs can be found on the GE website at https://digitalsupport.ge.com/en_US/Alert/GE-Security-Advisories. Customers can also sign up for notification of new SIMs and security advisories on the website.

Patching Third-Party Software

GE also recommends that customers keep operating systems, databases, and other third-party software in their environment up-to-date with the latest security patches from the software vendor. GE regularly validates the compatibility of selected GE products with third-party operating system security patches. More information on this process can be found on the GE Support website at https://digitalsupport.ge.com/en_US/Alert/GE-Security-Advisories.

Platform Configuration and Hardening

GE recommends configuring operating systems, databases, and other platforms per vendor recommendations or industry standards.

The following organizations publish best practices, checklists, benchmarks, and other resources for securing systems:

- *Center for Internet Security*: <http://www.cisecurity.org>.
- *NIST*: <http://checklists.nist.gov>.
- *Microsoft*: <http://technet.microsoft.com/security/default.aspx>.

Contact Information

If you purchased this product through an Authorized Channel Partner, please contact the seller directly.

General Contact Information

Online Technical Support & GlobalCare:	https://digitalsupport.ge.com
Comments about our manuals and online Help:	doc@ge.com
Additional information:	https://www.ge.com/digital/products

Technical Support

If you have technical problems that cannot be resolved with the information in this guide, please contact us by telephone or e-mail, or on the web at <https://digitalsupport.ge.com>.

Americas

Online Technical Support:	https://digitalsupport.ge.com
Telephone:	1-800-433-2682 1-617-725-2696 (if toll free 800 option is unavailable)
Technical Support e-mail:	digitalsupport@ge.com
Order Services (formerly Customer Care) e-mail:	GEDClientServices@ge.com
Sales Support (formerly Inside Sales) e-mail:	GEDigitalSalesSupport@ge.com
Primary language of support:	English

Europe, Middle East, and Africa

Online Technical Support:	https://digitalsupport.ge.com
Telephone:	+800 1-433-2682 +420-296-183-331 (if toll free 800 option is unavailable or dialing from a mobile telephone)
Technical Support e-mail:	digitalsupport.emea@ge.com
Order Services (formerly Customer Care) e-mail:	GEDClientServices@ge.com
Sales Support (formerly Inside Sales) e-mail:	GEDigitalSalesSupport@ge.com
Primary language(s) of support:	English, French, German, Italian, Czech, Spanish

Asia Pacific

Online Technical Support & GlobalCare:	https://digitalsupport.ge.com
Telephone:	+86-400-842-4213 +65-3157-4914 (India, Indonesia and Pakistan)
Technical Support e-mail:	digitalsupport.apac@ge.com
Order Services (formerly Customer Care) e-mail:	GEDClientServices@ge.com
Sales Support (formerly Inside Sales) e-mail:	GEDigitalSalesSupport@ge.com
Primary language(s) of support:	Chinese, English

Chapter 3. Tutorials

Tutorials

Configure access to specific equipment instance details

You can configure your environment to allow members of certain personnel groups to view the details of only specific equipment instances or levels of the equipment hierarchy from the **S95 Model Editor**.

About this task

- When group members have permissions on an equipment instance, they also have the equivalent permission on its child instances.
- When group members try to access levels of the equipment hierarchy that they do not have the **Read Equipment** permission for, a message is displayed in the **S95 Model Editor** stating that the user does not have permission to view the details of that equipment.



Note:

Any changes you make to a group's key sets take effect the next time the members of the group log in.

Procedure

1. If you are configuring a new key set, start here. If you are using an existing key set, skip to step 2.
 - a. In the navigator, click **Personnel > Key Sets > S95 Configurator**, and then select the new key set.
 - b. In the **Permissions** list, in the **Equipment Model Permissions** group, select the **Read Equipment** check box.



Note:

You can also provide the group members with the ability to add, update, and delete equipment instances.

- c. Click **Save**.
2. In the navigator, click **Personnel > Groups**, and then select the group you want to configure.
 3. In the **Displays** panel, click **Security Configuration > Security Editor**.
 4. Click **Assign Key Sets**.
 5. In the **Select a Group** area, select the group that you want to configure.

6. In the **Permission Details For** area, select the Edit by Key Set option.
7. In the **Edit by Key Set** area, under the **System Visibility** key set:

Click...	To...
Add Workstation	Add a specific workstation to the key set for the selected group.
Add All Workstations	Add all workstations in the system to the key set for the selected group.

8. Click **Save**.
9. In the navigator, select **Equipment > Instances**, and then navigate to and select the equipment instance that you want to provide access to.



Note:

You can configure access for multiple instances and at various levels; for example, you can configure access to a single production line at one site and all levels of the hierarchy at another site.

10. In the **Displays** panel, click **Security Configuration > Security Editor**.
11. In the **Select a Group** area, select the group you are configuring equipment access for.
12. In the **Permission Details For** area, select the Edit by Key Set option.
13. In the **Edit by Key Set** area, under the **Equipment Configurator** key set or another key set that you created for this purpose:

Click...	To...
Add Workstation	Add a specific workstation to the key set for the selected group.
Add All Workstations	Add all workstations in the system to the key set for the selected group.

14. Click **Save**.



Note:

The group members can access the selected instance and its child instances.

Developing a web form for a workflow

This tutorial shows how you can design, develop, and use a web-based form, then add it to a simple workflow for use by an operator working in the Task List.

Before you begin

The following procedures indicate the basic steps involved in creating a simple workflow with a web form.

- Log in and launch Workflow client
- Create and configure a web form with appropriate labels and operator controls



Tip:

Text fields can be used to display read-only information, such as headings or titles on your form. Text boxes can be used for processing operator input.

- Create and configure a workflow
- Add a form to a workflow
- Add activities to a workflow to demonstrate binding user input to a form
- Add instructions for operators
- Create a schedule to automatically start a workflow
- Test the workflow in the Task List

About this task



Note:

For a video tutorial of this topic, use a browser to open <http://www.youtube.com>, and then search for GE Proficiency Workflow - Developing a Web Form for a Workflow.

Procedure

1. Log in to Workflow.
 - a. Launch Workflow client using either the **Start** menu or desktop icon.
 - b. On the main Workflow screen, click  .
The **Workflow Login** dialog box appears.
 - c. **Optional:** In the **Authentication Type** section, select the type appropriate to your user profile.
 - d. In the **User Name** field, enter your user name.

- e. In the **Password** field, enter your password.
 - f. Click **OK**.
2. Create a web form resource.
 - a. From the navigator, click **Global Displays > Forms**.
 - b. Click **Add Form**.
 - c. In the **Name** field, enter a name, such as `My First Form`.
 - d. In the **Description** field, enter clarifying information about the web form resource, such as `This is my first web form`.
 - e. Click **OK**, or press `ENTER`.
 - f. In the **Displays** panel, click **Display Editor**.
 - g. On the **Overview** tab, select **Use the Designer tab to create a simple web form**.
 - h. Click **Save**.
 3. Design a form by adding a read-only text field to the web form.
 - a. Click the **Designer** tab.
 - b. In the **Form Fields** section, click the **Text** form control to add the field to your form.

**Tip:**

You can add multiple text fields, as well as other form controls. Each text control is labelled and numbered in the **Name** field in sequence; for example, `Text1`, `Text2`, and so on.

To delete a field, select it, and then click **Delete**.

The **Layout** area displays the basic configuration for your form.

- c. In the **Properties** section, you can configure the form fields.
 - i. In the **Name** field, enter a unique identifier by modifying the existing text. For example, enter `Text Control Name`. For example, enter `Text Control Name`.
 - ii. In the **Text** field, enter content that is displayed to the operator and provides more information about the form. For example, enter `Form Title`.
 - iii. In the **Size** field, from the drop-down list, select a text size.
 - iv. In the **Alignment** field, from the drop-down list, select an alignment value.
4. Design a form by creating a text box to the web form.

- a. In the **Form Fields** section, click the **Text Box** form control to add the field to your form. The **Layout** area displays the basic configuration for your form.
 - b. In the **Properties** section, you can configure the form fields.
 - i. In the **Name** field, enter a unique identifier by modifying the existing text. For example, enter `Text Box Control Name`.
 - ii. In the **Label** field, enter descriptive text on the form control that is displayed to the operator. For example, enter `Text Box with Status Message`.
 - iii. From the **Type** drop-down list, ensure **Text** is selected.

No further configuration is required.
 - c. Click **Save**.
5. To view your form or display, in the **Displays** panel, click **Global Viewer**.
 6. Create a workflow resource.
 - a. In the navigator, click **Workflow > Workflows**.
 - b. Click **Add Workflow**.
 - c. In the **Name** field, enter a name, such as `My First Workflow`.
 - d. In the **Description** field, enter clarifying information about the workflow resource, such as `This is my first workflow`.
 - e. Click **OK**, or press `ENTER`.
 7. Configure the workflow resource.
 - a. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
 - b. Click the **Workflow Editor** tab.
 - c. To assign a task in the workflow, on the **General** tab, in the **Personnel** field, click the ellipsis button.
 - d. In the **Universal Browser**, click **Personnel > People**, and then select your user name.
 - e. Click **OK**, or press `ENTER`.
 - f. Drag and drop a **Local Subprocess** into the workflow and double-click it, or double-click an existing subprocess.
 - g. From the subprocess configuration panel, click **Edit**.
 - h. In the **Name** field, enter a name, such as `WebFormStep`, and then click **OK**.
 - i. Click **Save**.
 - j. Click the **Overview** tab.
 - k. In the **Status** section, click **Enable**.
 8. Add a form to the workflow.

- a. Click the **Workflow Editor** tab.
 - b. From the workspace, double-click the subprocess, `WebFormStep`.
 - c. From the **Standard Activities** panel, drag and drop a **Form** activity into the subprocess.
 - d. From the activity configuration panel, click **Edit**.
 - e. In the **Name** field, enter a name, such as `WebFormActivity`, and then click **OK**.
 - f. From the activity configuration panel, in **General Settings**, click the ellipsis button next to the **Forms** field.
The **Universal Browser** appears.
 - g. In the **Universal Browser**, click **Global Displays > Forms**, and then select the form that you created, `My First Form`.
 - h. Click **OK**, or press `ENTER`.
 - i. Click **Save**.
9. Add activities to the subprocess to display form information to and request input from an operator. You can bind user input from the form's text box control (step 4) to workflow variables. For example, you can add a **Status Message** activity to your subprocess in order to deliver a message to your operator regarding a particular task, and request input.
- a. From the **Standard Activities** panel, drag and drop a **Status Message** activity into the subprocess after the **Form** activity.
 - b. From the activity configuration panel, in **General Settings**, click the drop-down arrow next to the **Message** field, and then select **Expression** to bind the message in your form.
 - c. In the **Enter Expression** field, enter the following expression: `"User entered:"+`
 - d. Click the **Workflow Variables** tab.
 - e. Expand the **Form** activity, `WebFormActivity`, that you added to the subprocess.
 - f. To complete the expression, double-click the text box value, `Text Box Control Name.Value`, that you created in step 4.
You have created the following expression: `"User entered:"+{WebFormActivity}.{Text Box Control Name.Value}`

- g. Click **OK**.
 - h. From the **Standard Activities** panel, drag and drop a **Delay** activity into the subprocess after the **Status Message** activity.
 - i. In the **Timeout** field, select the **seconds** section, and then use the arrows to set a 5-second delay.
 - j. Click **Save**.
10. Add work instructions to the workflow.
- a. Click the **Work Instructions** tab.
 - b. From the **Task/Task Steps** list, select the task step, `WebFormStep`.
 - c. In the **Work Instructions** area, enter instructions or insert an image.
 - d. Click **Save**.
11. Create a workflow schedule to start the workflow automatically, and then activate it so that it is available in production.
- a. In the navigator, click **Workflow > Schedules**.
 - b. Click **Add Schedule**.
 - c. In the **Name** field, enter a name, such as `My Schedule`.
 - d. In the **Description** field, enter clarifying information about the resource, such as `This is my first schedule`.
 - e. Click **OK**, or press `ENTER`.
 - f. In the **Displays** panel, click **Workflow Management > Workflow Editor > Overview**.
 - g. In the **Select Workflow** section, click **Browse**.
The **Universal Browser** appears.
 - h. Select the workflow that you created in step 6, `My First Workflow`
 - i. In the **Start Task Filter** section, select the **Is schedule visible in Start Task window** check box.
 - j. Click **Save**.
 - k. In the **Status** section, click **Enable**.

12. Run and test the workflow.

- a. From the title bar, click the Task List icon,  .
- b. Click **Start Task**.
- c. Select the schedule that you created in step 11, `My Schedule`.
- d. Click **Start**.

The task that you created is now visible in the Task List. You can enter data into the form, view work instructions, and submit changes.

What to do next



Tip:

For Vision users, click the **TASKS** tab to see pending workflows.

Creating unique displays, solutions, models

You can create displays, solutions, and models that are visible to specific users only.

About this task

To begin, you will create the following demonstration resources: User Joe; User Group; Display 1; Display 2; Display 3; Solution 1; Solution 2; Equipment Enterprise

Procedure

1. Create personnel resources, such as a user and a group of users.
 - a. In the navigator, click **Personnel > People**.
 - b. Click **Add User**.
 - c. In the **Name** field, enter a person's name, such as `User Joe`.
 - d. **Optional:** In the **Description** field, enter clarifying information for this resource.
 - e. Click **OK**, or press `ENTER`.
 - f. In the navigator, click **Personnel > Groups**.
 - g. Click **Add Group**.
 - h. In the **Name** field, enter a group name, such as `User Group`.
 - i. **Optional:** In the **Description** field, enter clarifying information for this resource.
 - j. Click **OK**, or press `ENTER`.
2. Activate the security credentials for `User Joe`.

- a. In the navigator, click **Personnel > People**.
- b. From the list of people, select the user, `User Joe`.
- c. In the **Displays** panel, click **S95 Model Editor**.
- d. Click the **Details** tab.
- e. In the **Full Name** field, enter `User Joe`, and then click **Create Account**.
- f. In the **Login Name** field, enter a unique user name for the user you are configuring.
The **Credentials** area expands, displaying further input fields.
- g. In the **Password** field, enter the password for the user you are configuring.
- h. In the **Confirm Password** field, re-enter the password.
- i. **Optional:** In the **Email** field, enter the e-mail address of the user you are configuring.
- j. Click **Save**.
The security credentials for `User Joe` are now active.

3. Add a user to a group of users.

- a. In the navigator, click **Personnel > People**.
- b. From the list of people, select the user, `User Joe`.
- c. In the **Displays** panel, click **Security Editor**.
- d. In the **Security Groups** area, click **Add/Remove Groups**.
The **Add/Remove Groups** dialog box appears.
- e. Select `User Group`, click , and then click **OK**.

4. Create global display resources.

**Note:**

To configure displays, see the **Forms and User Displays Authoring Guide**.

- a. In the navigator, click **Global Displays > Displays**.
- b. Click **Add Display**.

- c. In the **Name** field, enter a display name, such as `Display 1`.
 - d. **Optional:** In the **Description** field, enter clarifying information for this resource.
 - e. Click **OK**, or press `ENTER`.
 - f. Select `Display 1`, click **Duplicate**.
 - g. In the **Name** field, enter a new display name, such as `Display 2`.
 - h. Click **OK**, or press `ENTER`.
 - i. Again, select `Display 1`, click **Duplicate**.
 - j. In the **Name** field, enter a new display name, such as `Display 3`.
 - k. Click **OK**, or press `ENTER`.
 - l. In the navigator, click **Global Displays > User Solutions**.
 - m. Click **Add Solution**.
 - n. In the **Name** field, enter a display name, such as `Solution 1`.
 - o. **Optional:** In the **Description** field, enter clarifying information for this resource.
 - p. Click **OK**, or press `ENTER`.
 - q. Select `Solution 1`, click **Duplicate**.
 - r. In the **Name** field, enter a new user solution name, such as `Solution 2`.
 - s. Click **OK**, or press `ENTER`.
5. Create an equipment resource.
- a. In the navigator, click **Equipment > Instances**.
 - b. Click **Add Enterprise**.
 - c. In the **Name** field, enter a new display name, such as `Equipment Enterprise`.
 - d. Click **OK**, or press `ENTER`.
6. Add your new display resources to the user solution resources in order to group them.
- a. In the navigator, click **Global Displays > Displays**.
 - b. Select `Display 1`.
The **Display Editor** appears.
 - c. Click the **Display/View Config** tab.
 - d. In the **Display Configuration** section, select the **This display is available in the solution panel to the left** option.
 - e. In the **Display-Solution Configuration** section, select the user solution, `Solution 1`.
 - f. In the **Context View Configuration** section, select the associated check box.

- g. From the **Object Types** list, select any resource, such as **Equipment Resource**, to ensure that the selected solution is compatible and visible when the user selects any resource from the equipment model, and then click .
 - h. In the **Display-Solution Configuration** section, select the user solution, *Solution 2*.
 - i. Click **Save**.
 - j. Repeat steps a-i for the following resources: *Display 2* and *Display 3*.
7. Assign permissions to allow users in your group, *User Group*, to view particular displays.
 - a. In the navigator, click **Global Displays > Displays**.
 - b. Select *Display 1*.
 - c. In the **Displays** panel, click **Security Editor**.
 - d. In the **Select a Group** section, select **User Group**.
 - e. In the **Permission Details** section, scroll down to **System Visibility**, and then click **Add All Workstations**.
 - f. Click **Save**.
8. Assign permissions to allow users in your group, *User Group*, to view particular models, such as *Equipment*.
 - a. In the navigator, click **Proficy System > Proficy System > Models > Equipment**.
 - b. In the **Displays** panel, click **Security Editor**.
 - c. In the **Select a Group** section, select **User Group**.
 - d. In the **Permission Details** section, scroll down to **System Visibility**, and then click **Add All Workstations**.
 - e. Click **Save**.
9. Assign permissions to allow users in your group, *User Group*, to have access to a commonly used display.
 - a. In the navigator, click **Proficy System > Proficy System > Solutions > Production Management > Inventory View**.
 - b. In the **Displays** panel, click **Security Editor**.
 - c. In the **Select a Group** section, select **User Group**.
 - d. In the **Permission Details** section, scroll down to **System Visibility**, and then click **Add All Workstations**.
 - e. Click **Save**.

What to do next

When your user, `User Joe`, logs in to Workflow, he will see only the displays that were configured to be visible to him.

Create a work segment response for a work request using a workflow

Before you begin

In a workflow subprocess, you use two call method activities to perform this task. You can create expressions or bind properties for many of these properties, depending on your workflow requirements.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow. See **Add a Resource to the Workflow Model**.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IProductionRuntime** to display the list of available call methods.
 8. Click **CreateWorkResponse**.
 9.
 - a. Click **Next** to select a work type, and then click **Finish**.
The input parameters appear in the activity configuration panel.

- b. In the **Description** field, enter a description or expression, or bind a property or parameter to the method.
 - c. In the **Location** field, click the ellipsis button, and then select an equipment location.
 - d. In the **S95 ID** field, enter a name or expression, or bind a property or parameter to the method.
10. In the **Work Request** field, click the ellipsis button.
The **Universal Browser** appears.
11. Select the work request you want to associate with the work response.
12. Repeat steps 5-7.
13. Click **CreateSegmentResponse**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
14. Optionally:
- a. In the **Description** field, enter a description or expression, or bind a property or parameter to the method.
 - b. In the **End Time** and **Start Time** fields, select a date and time, or add an expression, or bind a property or parameter to the method.
 - c. In the **Location** field, click the ellipsis button, and then select an equipment location.
 - d. In the **S95 ID** field, enter a name or expression, or bind a property or parameter to the method.
15. In the **Work Process Segment** field, click the ellipsis button, and then select a process segment to associate with this segment response.
16. In the **Parent ID** field, click the arrow, and then select **Bind**.
The **Select Variable** dialog box appears.
17. Expand the CreateWorkResponse call method added in step 8, select a ReturnValue property, and then click **OK**.
18. Click **Save**.
19. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.
When the workflow runs, the segment response is created and can be viewed in the Response Information tab of the work request associated with the work request selected in step 11.

Write work data for a work segment response using a workflow

You can create expressions or bind properties for many of these properties, depending on your workflow requirements.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IProductionRuntime** to display the list of available call methods.



Note:

If you selected the **Required Class** check box on the **Work Data** tab of the associated work process segment, then go to step 8. If not, go to step 9.

8. Click **AddWorkDataClassToResponse**, and then click **OK**.
 - a. In the **Segment Response ID** field, click the arrow, and then select **Bind**.
The **Select Variable** dialog box appears.
 - b. From the **Activities** tab, select a **ReturnValue** for the segment response, and then click **OK**.
 - c. If required, in **Work Data Class**, bind a property the resource.
9. Click **WriteWorkData**.

- a. If required, to configure the data properties to write, click **Next**.
 - b. Click **Add classes of work data properties**, and then select a work data class.
The properties for the selected work data class appear.
 - c. To change properties, click **Remove classes of work data properties**, and then select a different work data class.
 - d. Click **Select all** or select the check box of a specific property, and then click **Finish**. Each property you select appears as an input parameter on the activity configuration panel.
 - e. For each property, enter a new value or expression, or bind a property or parameter to the method.
 - f. In the **Segment Response ID** field, click the arrow, and then select **Bind**.
The **Select Variable** dialog box appears.
 - g. From the **Activities** tab, select a **ReturnValue** for the segment response, and then click **OK**.
10. Click **Save**.
 11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

Configure forms to view during run time after they have been completed or acquired

Depending on your access permissions and client configuration, you can select either My Task List or Task Management view.

About this task

You can view completed or acquired forms in the Task List if they are configured to support this.

Procedure

1. Use the following table to perform tasks.

To view...	Go to step...
task history of a completed form	2
recently completed tasks during runtime	8

2. In the navigator, click **Workflow**.

3. In the **Folders** list, add a workflow or select an existing workflow. For more information, see **Add a resource to the workflow model**.
4. Click the **Tracking Configuration** tab.
5. Select **Enable Tracking**.
6. Select **Track at the Process and Subprocess Level**.
7. Click **Save**.
8. In the **Models** list, click **Global Displays**.
9. In the **Folders** list, add a form or select an existing form. For more information, see **Add a Global Displays resource**.
10. In the displays panel, click **Display EditorDesignerDocument TreeProperties**.
11. Select the **ReloadableForm** check box.
12. Optionally, to display read-only data in a completed or acquired (reloaded) state, use the *IsReloaded* variable in your form. This Boolean value indicates whether a form is being viewed for the first time (False), or if it has been reloaded (True).



Note:

You cannot submit a form that is loaded in this state.

The following is an example of a common purpose for using *IsReloaded*:

- a. From the Document Tree FormBase, select a user interface element that you want to view in the Task List.
 - b. From the Properties tab, click the **IsReadOnly** check box, and then from the drop-down arrow, select **Expression**. The Expression dialog box appears.
 - c. Click the **Expression Variables** tab.
 - d. Expand Controls, expand FormBase, and then double-click **IsReloaded** to add it to the expression field.
13. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms.
title bar	to the server for WPF forms.

14. You can view forms in several ways. From the Task List of a running workflow:

- Click the **Task History** tab, and then click the **Input** button. Recently completed task information is shown. To view previously run forms, from the Form drop-down menu, select a form.
- Click the **Tasks** tab, and then from the **Task Steps** list, select a previously completed step of a running task.
- You can view a currently running form that has been acquired by someone else.

**Note:**

A completed task cannot be re-executed; therefore, the **Submit** and **Cancel** buttons are not available.

Configure a database statement for a SQL service provider

After creating a database connection, you can configure database statement for a SQL service provider.

Procedure

1. In the navigator, click **Proficy System > Proficy System > Databases > Database Statement**.
2. Click **Add Statement**.
3. In the **Name** and **Description** fields, enter a name and description for the resource.
4. In the Displays panel, click **Database Editor**.
5. Click the **Designer** tab.
6. From the **Statement Type** drop-down list, select one of the following actions for a statement.

Select...	To...
Query	select
Update	insert, update, or delete

7. From the **Provider** drop-down list, select **SQL Data Provider**.
8. In the **Statement** section, enter a valid SQL string, and then click **Create Parameters** to generate automatically any parameters used in the statement.
9. **Optional:** To add other parameters, in the data grid, click **Add**.
10. **Optional:** To configure the data grid properties:
 - In the **Name** column, select a cell, and then enter a new name.
 - In the **Data Type** column, select a cell, and then from the drop-down list, select a data type.

**Note:**

The parameters must match those used in the statement string.

11. To remove a parameter, in the data grid, select a row, and then click **Delete**.
12. Click **Save**.
13. To test the statement:
 - a. Click the **Test** tab.
 - b. In the **Inputs** section, in the **Parameter Value** field, enter a value for each parameter.
 - c. In the **Connection** section, click **Browse**.
The **Universal Browser** appears.
 - d. Select a database connection to run the statement against, and then click **OK**.
 - e. In the **Test** section, click the **Test** button.



CAUTION:

Statements are executed against a specified connection, which may result in loss of data.

The **Outputs** section displays the test results.

14. To execute a statement, open a workflow, and then insert and configure a Call Method activity.
15. From the call method activity configuration panel:
 - a. Click **Configure Method**.
 - b. Expand **IDatabaseService**, select **ExecuteStatement**, and then click **Next**.
 - c. In the **Statement** section, click **Browse** next to the **Use Existing** radio button.
The **Universal Browser** appears.
 - d. Select a statement, and then click **Finish**.
 - e. On the call method activity configuration panel, in the **Connection** field, click the ellipsis button.
The **Universal Browser** appears.
 - f. Select a database connection, and then click **OK**.
 - g. To bind the connection to a workflow or subprocess property, click the drop-down arrow, and then select **Bind**.

- h. From the Call Method activity panel, in the **Inputs** section, you can configure any required input parameters.

**Note:**

The **Statement** field is read only.

- i. Click **Save**.

16. **Optional:** You can also execute a procedure from WPF and Silverlight forms by invoking a server method. For more information see **Bind a return value to a DataGrid using from server methods**.

Configure multiple servers on a single machine

After installing application server, you can configure a multiple server environment onto a single machine. This procedure demonstrates how to create two additional server instances; however, you can create as many instances as you require.

Before you begin

Install a single application server and client.

**Important:**

- You must exit all instances of Workflow that are located on the server that you are working on.
- You must stop all running workflows on your system before creating a new server instance and corresponding services.
- For each server instance, use a unique server name (see step 2b) and port number (see step 2f).

About this task

You can configure multiple servers on a single machine for two main reasons.

- If your SOAServer must run on a 32-bit system and is approaching 3 GB of memory usage, then you can move workflows or service providers to a separate server instance to reduce memory usage on the SOAServer.
- If custom service providers are unstable and you must restart SOAServer frequently, you can move customer service providers to separate server instances.

Procedure

1. Open the **Services** window to stop all services.
 - a. From the **Start** menu, go to **Control Panel > Administrative Tools > Services**.
 - b. In the **Name** column, click **Stop the service** for each of the following services: Proficy Server, Proficy STS, Proficy Publisher Service.
2. Create a new server instance using the **Configure Server** tool.
 - a. On the server machine, On the server machine, click **Start > All Programs > General Electric > Workflow > Configuration > Configure Server**.
 - b. In the **Name** field, enter a name for the new server instance, such as `WorkflowServer`.
 - c. In the **Description** field, enter a description for the server instance.
 - d. Click the **Advanced Configuration** arrow.
 - e. Select **64 bit** to run the server using that architecture.



Note:

We recommend that you use a 64-bit system unless your service provider requires 32-bit.

- f. In the **P2P Configuration** section:
 - In the **Port** field, enter the port number for the server instance. This value must be unique if multiple server instances are run on the same machine.



Note:

For the **IP Address** field: Peer servers will contact this server instance at the specified IP address. If the specified address is `255.255.255.255`, then the address will be determined at server start as the first IPv4 address found in the DNS records for the hosting machine's host name. If the address is incorrect, then an appropriate address should be specified here.

- g. Click **Save**.
The following service has now been created: Proficy Server - Workflow Server.
- h. Click **Back**.
- i. In the **Name** field, enter a new name for `SOAServer`, such as `UserServer32`.

- j. In the **Description** field, enter a description for the server instance.
- k. Click the **Advanced Configuration** arrow.
- l. Select **32 bit** to run the server using that architecture.
- m. In the **Port** field, enter the port number for the server instance. This value must be unique if multiple server instances are run on the same machine.
- n. Click **Save**.

**Important:**

After saving the entered data, port numbers are validated to ensure that the same one is not used more than once. To find a new port number, enter a number that is close to the existing one. If a port is already in use, an error will occur. You are then required to return to the entry fields and enter all data again.

The following service has now been created: Proficy Server - UserServer32.

- 3. Move service providers to the new server instances using the **Configure Site** tool.
 - a. On the server computer, click **Start > All Programs > General Electric > Workflow > Configuration > Configure Site**.
The **Configure Site** tool appears.
 - b. In the **Application Server Instances and Enabled Product Options** pane, select the service provider from `SOAServer` that you want to transfer to the new instances, and then drag and drop it below the new host servers, `WorkflowServer` and `UserServer32`. Repeat this process until all service providers have been moved.
 - c. Click **Save**.
- 4. Open the **Services** window to start each service again.
 - a. In the **Name** column, click **Start the service** to start each of the following services again:
Proficy Server, Proficy STS, Proficy Publisher Service, Proficy Server - UserServer32, Proficy Server - WorkflowServer.
- 5. Launch the Workflow client using either the **Start** menu or desktop icon.

Chapter 4. Orientation

Workflow Overview

Orientation

This product is a secure framework for applications that are hosted within it, and it provides a common set of models the applications can use. These models interact with a set of data models that can be configured for operational reference and historical data storage. They also reference a variety of data within a model during execution.

Multiple Sessions of Workflow

In some situations, multiple users may need to access Workflow from the same work station to perform tasks specific to their respective responsibilities.

You can use the **New Session** button to initiate a new Workflow instance, as well as navigate between multiple instances using the session drop-down list.



Note:

To log in to Workflow, the system must be in a state of either Complete or Partial health. For more information, see **System Health and Program Use**.

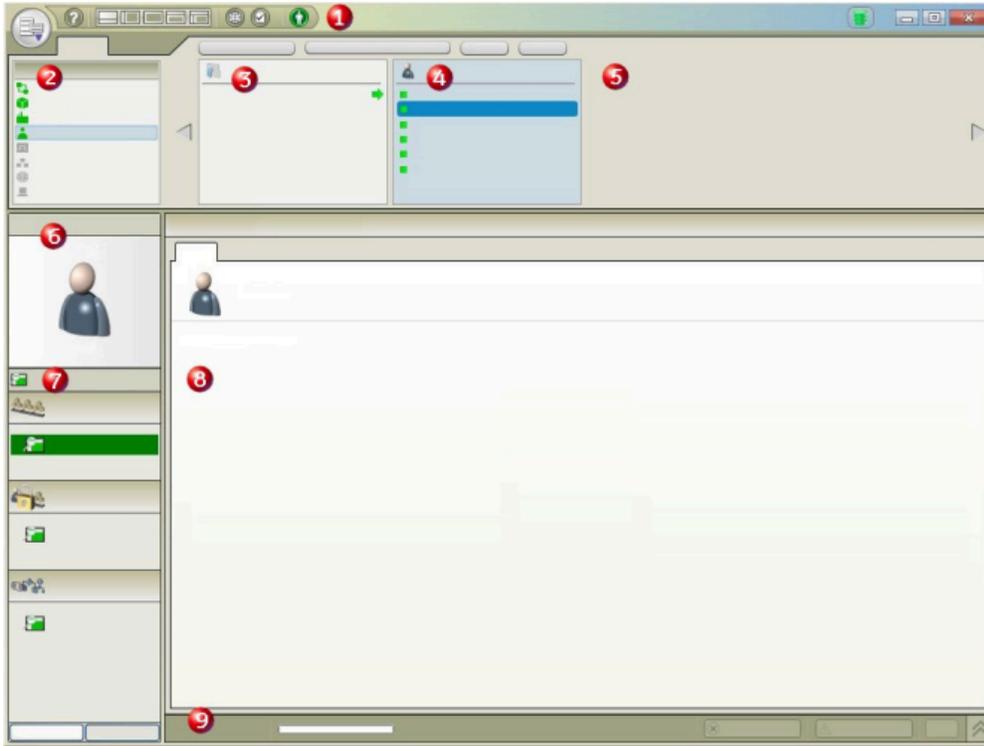
User Interface

User Interface

This document is a general guide to the layout of Workflow.

Overview

The diagram displays the main components of Workflow. The table following describes these components.



Cross Reference	Name	Description
1	Menu Bar	Includes: Help, layout modes, Global Displays, Task List, login/logout, site health status
2	Models	Includes list of vsailable models to use for configuration.
3	Resource type list	Includes he types of resources that kyou can add a resource to.
4	Resources	Specific objects that you can configure.
5	Navigator	Allows you to locate and manipulate resources.
6	Compatible Views	Displays options that are compatible with the selected resource.
7	Displays panel	Selection of editors that allow you to configure resources.
8	Workspace	Area where you configure resources.
9	Information Bar	Includes system level information, such as error messages.

Menu Bar

The Menu Bar has various components to help you navigate the system.

Icon	Indicator	Description
	Main Menu	Application, system, and technical support information.

Login

Icon	Description
	Click to log into or out of Workflow.

Help Button

Icon	Description
	Click to access help files.

Global Viewer

Icon	Description
	Click to access the Global Viewer display in order to preview forms and displays.

Task List Indicators

Icon	Description
	Click to access the Task List.
	When this icon is green, it indicates that tasks are waiting to be worked on.
	Click to display expired tasks.

Frame Layout

When you select a particular icon, you change the layout of your operating environment.

Icon	Description
	Shows the navigator only.
	Shows the Displays panel and the main workspace only.
	Shows the main workspace only.
	Shows the navigator and the workspace.
	Shows all the features of your operating environment, such as the Displays panel, the navigator, and the workspace.

Site Health Indicators

Icon	Description
	<p>Indicates that site health is <i>Complete</i>: All servers, both essential and nonessential, are running and reachable, and all program functionality is available. For more information, see System Health and Program Use.</p> <div data-bbox="824 1394 1425 1663" style="border: 1px solid orange; border-radius: 10px; padding: 10px;"> <p>! Important: A user's ability to connect to the server also depends on security permissions. For more information about security features, see Workflow Security. Security.</p> </div>
	<p>Indicates that site health is one of the following:</p> <ul style="list-style-type: none"> • <i>Partial</i>: One or more nonessential servers are not running or are unreachable. In this scenario, active users remain logged in to

Icon	Description
	<p>the program, and inactive users can log in, but program functionality tied to product options hosted on compromised servers is unavailable.</p> <ul style="list-style-type: none"> • <i>Waiting</i>: One or more essential servers that do not host the Core product option are not running or are unreachable. In this scenario, active users are logged out of the program, and inactive users cannot log in. <p>For more information, see System Health and Program Use.</p>
	<p>Indicates that site health is <i>Unavailable</i>: The server that hosts the Core product option (typically, SOA Server) is not running or is unreachable. In this scenario, active users are logged out of the program, and inactive users cannot log in. For more information, see System Health and Program Use.</p>

Models

In Workflow, models are used to define and organize system and application information. A model is a representation of a physical structure or conceptual idea and is composed of resources that have hierarchical or other relationships between them. Rules or schemas outlining resource nesting, order, number, and requirement define the structure of a model.

Production and manufacturing models are used to represent your production or manufacturing organization and operation. Based on the industry standard ISA-95, these models define basic structure rules and object relationships.

Production models use properties that can define classes, definitions, and specifications. When you create your production or manufacturing structure and resource relationships, the properties, classes, definitions, and specifications are used to define your processes and can be inherited by another resource as a result of direct associations.

**Note:**

You can create up to 5000 resources for each model, such as 5000 material definitions, 5000 users, and 5000 work definitions.

Navigator

The navigator allows you to navigate models to target resources.

Using the navigator, you can add, delete, duplicate, modify, and browse resources within Workflow.

Universal Browser

The **Universal Browser** is a stand-alone version of the navigator that appears when selecting a specific sequence of options. You select resources that are used in the configuration of other resources.

Display Panel

Workflow uses displays to organize editors and resource information into relevant groupings that make it easy to select, view, and configure resources. You can view and configure model resources within Workflow.

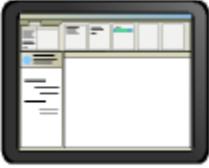
If you are working in a layout that exposes the Displays panel, you can maximize your working area by clicking the toggle arrow at the top of the panel, which alternately collapses or expands the panel with each click. In the collapsed mode, the panel appears as a thin vertical bar. Clicking on the bar causes the Displays panel to open in pop-up mode, from which you can select a different display. After selecting a new display, moving your cursor off the pop-up panel collapses it again.

**Note:**

The pop-up version of the panel does not allow you to choose between compatible or all displays. If you need to do so, you must click the toggle arrow to expand the full panel.

Displays

The following table describes the icons associated with displays:

Icon	Description
	<p>The selected display is not related to a resource. It is always available regardless of which model is selected.</p>

Compatible Views

Displays are organized by operational function, and are available as a compatible view when you select an appropriate resource. This organization makes it easier for you to remember where you are within the model framework.

When you navigate to a resource, you are automatically shown the compatible displays of the resource you are viewing. If you navigate to a resource that does not have a compatible display, you cannot work with that resource. If you want to access a specific display, you must have added a resource to the folder that it applies to. Then, the display is enabled to further select, view, or configure a resource.



Note:

The enabled displays available depend on both Workflow and any corresponding modules or Workflow compatible products you have installed. If a new product is installed but does not appear in the available views, close and re-open the Workflow client and check again.

Compatible Resources

The following table describes the icons associated with resources:

Icon	Indicator	Description
	solid green square	Resource works with the current display.
	solid green arrow	Resource has children that work with the current display.
	gray arrow	Resource has children, but none of them work with the current selection.

Compatible Views

The following table describes the icons associated with views:

Icon	Indicator
	Selected resource has a compatible view.
	Selected resource does not have a compatible view.

Editors

The editors allow you to design and develop all aspects involved in your production process.

Each model relates to specific resources that provide you access to an editor. When you select a resource related to a model, you use an editor to configure and modify it.

Resources in Workflow are organized by model; a model is a collection of related resources. Some editors work with all editable resources in a model, and some editors work with editable resources across all models. If a resource is read-only, there is no editor associated with it. Some resources have multiple editors that can configure different aspects of a resource. For example, an equipment resource's basic properties can be edited using the S95 Model Editor, and its security settings can be edited using the Security Editor.

To locate an editor for a specific resource, click the resource in the navigator. The icon for the applicable editor in the solution panel turns green.

Feedback Zone

The Feedback Zone is a tool that displays error and informational messages about the currently executing process or the operation last performed. Various components that are hosted within Workflow can also post messages about the results of operations and tasks in the Feedback Zone.

The Feedback Zone is a dynamically hidden pane at the bottom of the Workflow client workspace. You can open the Feedback Zone to view and/or clear messages. Different types of messages are

displayed within the Feedback Zone, including confirmations, information, warnings, errors and debugging information.

The Feedback Zone also provides you with status information about the currently executing or the last performed operation. Various components that are hosted within Workflow client can post messages about the results of operations and tasks in the Feedback Zone. When a new error is posted to the Feedback Zone, you are notified by an animated Error box in the status bar. Additional detailed information is stored in log files within the application's directory structure for redundancy and history.

The types of data reporting in the Feedback Zone are:

- Status information for long-running operations
- Lists of successful and failed operations for complex actions
- Information in the status bar of the Feedback Zone shows the number of selected objects, the number of error and warning messages, progress bar, and status of the last result, whether it was successful or unsuccessful.

The Feedback Zone is available on all Workflow client layouts.

Resources

Resources

In Workflow, a resource is a component of a model. A resource represents a real world entity or a physical item. To model your production organization, you define and apply meaningful properties to resources, providing a framework to describe your assets and processes.

Directory of Resources

The Directory of Resources (DOR) provides a distributed repository of all the resources that are created in Workflow. These resources include such things as customized user displays, alarms, reports, and data items within the server namespace. The DOR provides a directory of this information and the services that can be used to access the information.

There are two types of information published in the directory: standard system registrations and user published content. User published content provides complete programming examples of providers and consumers that publish information to the directory. For example, user published content includes the content of projects, data sources, events, and the equipment model.

The Directory of Resources supports greater than 5000 child nodes.

AD LDS (Active Directory Lightweight Directory Service)

Active Directory Lightweight Directory Service (AD LDS) is a light-weight implementation of Active Directory.

AD LDS is capable of running as a service on computers running Microsoft Windows Server 2003. AD LDS shares its code base with Active Directory and provides the same functionality as Active Directory, including an identical API, but does not require the creation of domains or domain controllers.

Like Active Directory, AD LDS provides a Data Store, which is a hierarchical data store for storage of directory data, a Directory Service with an LDAP Directory Service Interface. Unlike Active Directory, however, multiple AD LDS instances can be run on the same server, with each instance having its own applications making use of the specific AD LDS directory service instance.

Search for Resources

Searching for Resources

In the navigator, you can quickly locate any resource that exists in your project without having to browse for them. You can use both a basic and advanced searches.

IDirectorySearch

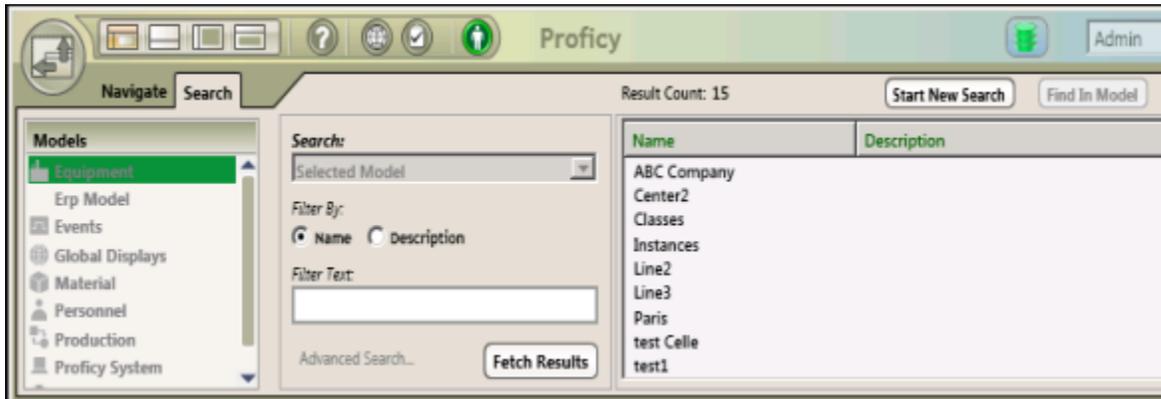
IDirectorySearch allows you to search the Directory of Resources for a set of resources based on a search query, or for a list of resources based on a resource class.

On the activity configuration panel, you can use a bind or expression to define your input and output parameters.

Subtopics in this section describe each service method for this service interface.

Basic Search

About this task



Procedure

1. In the Navigator, click the **Search** tab.
2. From the **Search** drop-down list, select a search criterion.
3. Optionally, in the **Filter By** section, click **Name** or **Description**.
4. Optionally, in the **Filter Text** section, enter text that matches resources you want to search for.
5. Click **Fetch Results**.

The search results are displayed in the **Results** pane.

6. Other basic search functions:

To...	Click...
perform another search	Start New Search
view your results in context	Find in Model
	Your search results appear in the navigator.

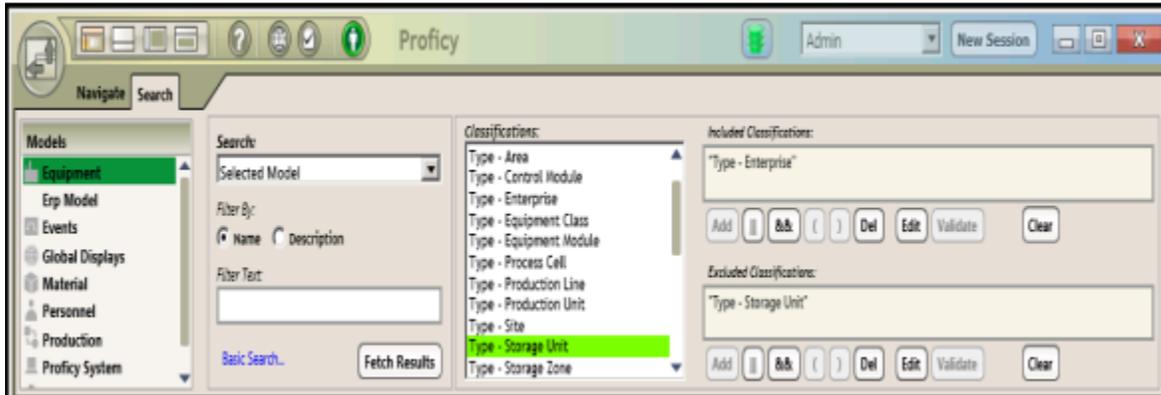
Related information

[Searching for Resources \(on page 245\)](#)

Advanced Search

With advanced search, you can quickly locate any resource that exists in your project by constructing search expressions. You create expressions using classifications, which consist of a model and an attribute or a property that is defined within that model.

Search Attributes



The search panel for the advanced search option has the following attributes:



Note:

If you include a text criterion in your search, the results match anything that includes the specified text as though it was a `*<text>*` wildcard search. For example, if you add a text expression of '1' to your search, any item with a '1' anywhere in its name (or description, if you search on that attribute) is shown as a search result.

Attribute	Value	Description
Hits count	Integer	Displays the maximum number of records that will be returned if no filtering is applied. This value changes as you add filters.
Search	System-defined	Select an option from the drop-down list to specify the search type. If you select All Models, the models menu is not available and all models are searched. If you select Selected Model, the models menu is enabled and you

Attribute	Value	Description
		<p>can select a specific model to search.</p> <p>If a resource is selected in the Navigate tab, additional options are available in the Search list. These options are displayed in blue text.</p>
Filter By: Name	N/A	Select this option to filter results by name.
Filter By: Description	N/A	Select this option to filter results by description.
Filter Text	System-defined alphanumeric string (A-Z, a-z, 0-9)	<p>Enter the text you want to filter by.</p> <div data-bbox="1024 953 1422 1129" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: Filtering is case sensitive. </div>
Classifications	System-defined	Select an option from the menu to include in your search expression.
Included Classifications	System-defined	Classification expression that defines which classifications will be included in the search.
Excluded Classifications	System-defined	Classification expression that defines which classifications will be excluded from the search.
Add button	N/A	Click to add a classification into the expression field.
Operator buttons:	N/A	Click to insert an "Or" symbol.
Operator buttons: &&	N/A	Click to insert an "And" symbol.

Attribute	Value	Description
Operator buttons: ()	N/A	Click to insert left and right parentheses.
Del button	N/A	Click to remove the last item inserted into the expression field.
Edit button	N/A	Click to edit the expression field and enable validation.
Validate button	N/A	Click to validate the expression. If the expression is not valid, the text changes to red.
Clear button	N/A	Click to clear the expression field.
Basic Search	N/A	Click this link to return to the basic search feature.
Fetch Results	N/A	Click this button to apply any filtering criteria and retrieve the search results.

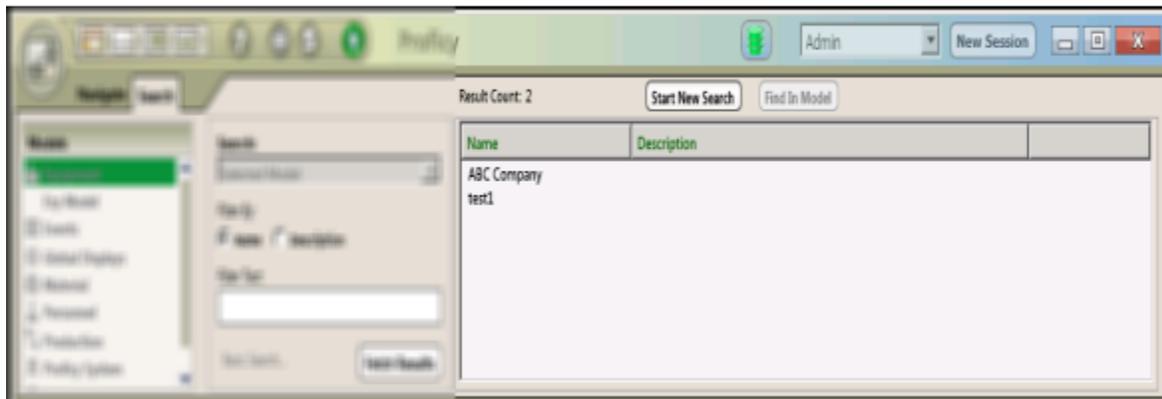
Search Results

The results of your search are displayed in the results panel of the Search tab when you execute the search.



Tip:

You can change the filtering criteria and click the Fetch Results button as often as needed to improve search results.



The results panel of the navigator has the following attributes:

Attribute	Value	Description
Results	Integer	Displays the total number of results for a search based on the criteria entered.
Start New Search	N/A	Click this button to remove results from the Results panel and return to the advanced search feature.
Find in Model	N/A	Click this button to locate and highlight the selected resource in the Navigate tab.
Name	N/A	Displays the name of any resources found during the search.
Description	N/A	Displays the description of any resources found during the search.

Expression Editor

Expression Editor

Expressions combine or manipulate one or more pieces of data into a new piece of data using mathematical or boolean logic, as well as a set of built-in functions; therefore, they must follow the standard algebraic rules for order of operations.

You can build simple or complex expressions. They are user-defined and consist of one or more of the following components:

- Functions
- Operators
- Variables
- Literals

The following table describes the general attributes that are available in all expression editor dialog boxes:

Attribute	Value	Description
Enter Expression	User-defined	Specifies an expression using the available functions, variables, values, and operators.
Operator buttons	System-defined	Click to insert an operator into your expression.
Check Syntax	System-defined	Click to confirm if an expression has been completed and ordered correctly.
Test	System-defined	Click to evaluate the expression. The Test Expression dialog box appears.
Test Expression dialog box	N/A	Click the Test button to open this dialog box. Click Evaluate to evaluate the expression. Or, if you use a predefined variable as part of your expression, enter the value here.

Attribute	Value	Description
Functions tab	N/A	Displays the functions available to build your expression. Double-click a function to insert it into your expression.
Variables tab	N/A	Displays the variables available to build your expression. Double-click a variable to insert it into your expression.
Workflow Variables tab	N/A	Displays the workflow variables available to build your expression. This tab is only available when you are in a workflow.

**Note:**

The property you are creating an expression for and its data type are displayed in the Expression Editor in the upper left and upper right, respectively.

Expression Functions

A function performs a calculation between values in order to produce another value. When functions are used in an expression, they aid in the expression's evaluation.

There are many function types that can be used when developing expressions. For more information, see:

- Expression Functions:
 - Math
 - Conversion
 - Logical
 - Text
 - Date
 - Time Span

Examples

- To check if the difference between two variables has exceeded a limit of 100:
 - $ABS(a - b) > 100$

- To calculate the maximum of two values:
 - IF (a >= b, a, b)
- Pythagorean theorem:
 - SQRT(POWER(a, 2) + POWER(b, 2)) > c

Related reference

[Math Functions in Expressions \(on page 253\)](#)

[Conversion Functions in Expressions \(on page 255\)](#)

[Text Functions in Expressions \(on page 257\)](#)

[Date Functions in Expressions \(on page 265\)](#)

[Time Span Functions in Expressions \(on page 281\)](#)

Math Functions in Expressions

Math functions are used for mathematical calculations in an expression.

Math

The following information describes the functions available for building expressions.

Function	Syntax	Description	Example
ABS	ABS(number)	Returns the absolute value of a number.	ABS(-5) = 5
AVERAGE	AVERAGE(number1, number2, ...)	Returns the average (arithmetic mean) of a group of numbers.	AVG(4, 10) = 7
ROUND	ROUND(number, [decimals])	Rounds a value to the nearest integer or specified number of decimal places.	ROUND(3.67) = 4
LN	LN(number)	Returns the natural logarithm of a number.	LN(E * E) = 2

Function	Syntax	Description	Example
LOG	LOG(number, base)	Returns the logarithm of a number to the base you specify.	LOG(5, 625) = 0.25
LOG10	LOG10(number)	Returns the base-10 logarithm of a number.	LOG10(1000) = 3
MAX	MAX(number1, number2, ...)	Returns the number representing the largest value among a group of numbers.	MAX(4, 2, 8, 1, 5) = 8
MIN	MIN(number1, number2, ...)	Returns the number representing the smallest value among a group of numbers	MIN(4, 2, 8, 1, 5) = 1
POWER	POWER(number, power)	Returns the result of a number raised to a power.	POWER(2.5, 3) = 15.625
EXP	EXP(number)	Returns E (the base of natural logarithms) raised to the specified power.	EXP(2) = 7.38905609893065
SQRT	SQRT(number)	Returns the square root of a number.	SQRT(64) = 8
SIN	SIN(angle)	Express angle in radians. Returns the sine of an angle.	SIN(PI / 6) = 0.5
COS	COS(angle)	Express angle in radians. Returns the cosine of an angle.	COS(PI / 3) = 0.5
TAN	TAN(angle)	Express angle in radians. Returns the tangent of an angle.	TAN(PI / 4) = 1

Function	Syntax	Description	Example
COT	COT(angle)	Express angle in radians. Returns the cotangent of an angle.	$COT(\pi / 2) = 0$
ACOS	ACOS(number)	Returns the arc-cosine of a number in radians.	$ACOS(0.5) = \pi / 3$
ACOT	ACOT(number)	Returns the arc-cotangent of a number in radians.	$ACOT(1) = \pi / 4$
ASIN	ASIN(number)	Returns the arcsine of a number in radians.	$ASIN(0.5) = \pi / 6$
ATAN	ATAN(number)	Returns the arctangent of a number in radians.	$ATAN(1) = \pi / 4$

Conversion Functions in Expressions

Conversion functions convert expressions of one data type to another data type when data type conversions are not automatically performed by SQL Server.

This function uses a date, time, or numeric value and returns a formatted string representing that value according to a specified style string.

You can find these formats at the following Microsoft websites:

- Predefined numeric formats: <http://msdn.microsoft.com/en-us/library/y006s0cz.aspx>
- User-defined numeric formats: <http://msdn.microsoft.com/en-us/library/4fb56f4y.aspx>
- Predefined date/time formats: <http://msdn.microsoft.com/en-us/library/az4se3k1.aspx>
- User-defined date/time formats: <http://msdn.microsoft.com/en-us/library/8kb3ddd4.aspx>

Conversion

The following information describes the functions available for building expressions.

Function	Syntax	Description	Example
Trunc	TRUNC(number)	Calculates the integral part of a number.	TRUNC(3.67) = 3
ToDouble	ToDouble(number)	Converts a number or string to a double.	"4.5" = 4.5
ToDecimal	ToDecimal(number)	Converts a number or string to a decimal.	"4.5" = 4.5
ToString	ToString(object)	Converts a value to a string. A null value becomes "NULL."	ToString(4.5) = "4.5"
Format	FORMAT(number, style)	Converts a number or date to a string formatted according to instructions contained in a format string expression. Returns a string value.	FORMAT(4.5, "E") = "4.500000E+000"

Logical Functions in Expressions

Logical functions perform an operation or comparison on objects and expressions.

Logical

The following information describes the functions available for building expressions.

Function	Syntax	Description	Example
If	IF(boolean_condition, expression1, expression2)	If the condition evaluates to True, expression1 is returned. If the condition evaluates to False, expression2 is returned.	IF (5>4, 1, 0) = 1



Note:
This function accepts any

Function	Syntax	Description	Example
		 type for its second and third parameters, as long as they are both of the same type.	
IsNull	IsNull(object)	Determines whether or not a given expression is type DBNull.	IsNull(myLocalVariable)

Text Functions in Expressions

Text functions perform an operation on a string input value and return a string or numeric value.

The following information describes the functions available for building expressions.



Tip:

- Any information within square brackets [] is optional.
- All indexing is one-based.
- For more information on formatting strings, see **Conversion Functions**.

StringLength

Description

Returns the number of characters in a string.

Format

StringLength(string)

Parameters	Required?	Data Type	Description
string	Yes	String	The string to perform the operation on.

LeftSubstring

Description

Returns the first (leftmost) characters from a string.

Format

LeftSubstring(string, nCount)

Parameters	Required?	Data Type	Description
string	Yes	String	The string to perform the operation on.
nCount	Yes	Numeric	Number of characters to retrieve.

RightSubstring

Description

Returns the end (rightmost) character from a string.

Format

RightSubstring(string, nCount)

Parameters	Required?	Data Type	Description
string	Yes	String	The string to perform the operation on.
nCount	Yes	Numeric	Number of characters to retrieve.

Substring

Description

Returns a substring of a specified length starting at the first specified character of a string.

Format

Substring(string, nFirst, nCount)

Parameters	Required?	Data Type	Description
string	Yes	String	The string to perform the operation on.

Parameters	Required?	Data Type	Description
nFirst	Yes	Numeric	First character of the string to retrieve (one-based index).
nCount	Yes	Numeric	Number of characters to retrieve.

Substitute

Description

Replaces all instances of one substring with another in a specified string, and returns the result as a string.

Format

Substitute(string, oldText, newText [, ignoreCase])

Parameters	Required?	Data Type	Description
string	Yes	String	Original string.
oldText	Yes	String	Substring to be replaced.
newText	Yes	String	Substring that overwrites the original text.
ignoreCase	No	Boolean	True if character case should be ignored when searching the string; otherwise, false.

FindString

Description

Searches a string (from left to right) to find the first occurrence of a given substring. Returns (Int 32) the character location of the first occurrence of the substring in the searched string (one-based index). Returns zero if the substring is not found.

Format

FindString(textToFind, textToSearch, startIndex [, ignoreCase])

Parameters	Required?	Data Type	Description
textToFind	Yes	String	Substring to search for.
textToSearch	Yes	String	String to search.
startIndex	Yes	Numeric	Location where the character search begins (one-based index).
ignoreCase	No	Boolean	True if the character case should be ignored when searching the string; otherwise, false.

ContainsString

Description

Determines whether a substring is present within a specified string, and returns a boolean result.

Format

ContainsString(textToFind, textToSearch [, ignoreCase])

Parameters	Required?	Data Type	Description
textToFind	Yes	String	Substring to search for.
textToSearch	Yes	String	String to search.
ignoreCase	No	Boolean	True if the character case should be ignored when searching the string; otherwise, false.

InsertString

Description

Inserts the given text into a string at a specified position, and returns the result as a string.

Format

InsertString(textToInsert, string, startPosition)

Parameters	Required?	Data Type	Description
textToInsert	Yes	String	String to insert
string	Yes	String	Original string.
startPosition	Yes	Numeric	Location where text is inserted into the original string (one-based index).

RemoveString

Description

Removes a portion of text from a string, and returns the result as a string.

Format

RemoveString(string, startPosition, characterCount)

Parameters	Required?	Data Type	Description
string	Yes	String	Original string.
startPosition	Yes	Numeric	First character of the string to remove (one-based index).
characterCount	Yes	Numeric	Number of characters to remove.

IsStringEqual

Description

Compares the text of two strings to determine if they are equal, and returns a boolean result.

Format

IsStringEqual(stringA, stringB [, ignoreCase])

Parameters	Required?	Data Type	Description
stringA	Yes	String	One of the strings to compare.
stringB	Yes	String	Other string to compare.

Parameters	Required?	Data Type	Description
ignoreCase	No	Boolean	True if the character case should be ignored when searching the string; otherwise, false is the default.

ApplyMask

Description

Tests a string to determine if it matches a specified pattern, and returns a boolean result.

For more information, see **Apply Mask Function**.

Format

ApplyMask(string, mask)

Parameters	Required?	Data Type	Description
string	Yes	String	Length of string to test.
mask	Yes	String	Mask string to apply.

ToLower

Description

Converts all uppercase letters in a text string to lowercase, and returns the result as a string.

Format

ToLower(string)

Parameters	Required?	Data Type	Description
string	Yes	String	String to convert to lowercase.

ToUpper

Description

Converts all lowercase letters in a text string to uppercase, and returns the result as a string.

Format

ToUpper(string)

Parameters	Required?	Data Type	Description
string	Yes	String	String to convert to uppercase.

TrimWhitespace

Description

Removes all occurrences of white space characters from the beginning and end of a string, and returns the result as a string.

Format

TrimWhitespace(string)

Parameters	Required?	Data Type	Description
string	Yes	String	String that the white-space is removed from.

Split

Description

Splits a string into an array of strings.

Format

string[] Split(stringToSplit [, delimiterString])

Parameters	Required?	Data Type	Description
stringToSplit	Yes	String	String that is converted into an array of strings.
delimiterString	No	String	Specifies the character that separates the sequence. It can be multiple characters.

Apply Mask Function

The apply mask function allows you to match a pattern in a string; that is, each character in a mask is tested against the corresponding character in the input string.

The mask is applied to a string on a character by character basis so that each character in the string is compared to its corresponding character in the mask, ensuring it is the correct type of character for that position, such as a digit, a letter, or a specific character, and so on. If the mask character used is specified as 'required', the character in the input string must be present and the character type must correspond to the mask character. If the mask character used is specified as 'optional', the character in the input string is valid against the mask if it is the expected character type or if it is left blank. A character is considered to be blank if it is a space (' ') or an underscore ('_').

The following table describes the effect of a character in a mask string:

Mask Character	Meaning	Description
0	required digit	Any digit.
9	optional digit	Any digit.
#	optional digit	Any digit; also matches the + and - characters.
L	required letter	Any letter.
?	optional letter	Any letter.
&	required character	Any non-control* character.
C	optional character	Any non-control* character.
A	required alphanumeric	Any letter or digit.
a	optional alphanumeric	Any letter or digit.
<	shift down	Flags mask to ensure that all letters that follow are lowercase, if matched by a mask character. This does not apply to literals.
>	shift up	Flags mask to ensure that all letters that follow are uppercase, if matched by a mask character. This does not apply to literals.
	cancel shift	Cancels any shift up or shift down effect. All letters that follow may be either upper or lowercase.

Mask Character	Meaning	Description
\	escape	Escapes a mask character, turning it into a literal. <div data-bbox="1024 390 1419 653" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;"> <p> Tip: You can use this feature on the following characters only: 0, 9, #, L, ?, &, C, A, a, <, >, , or \</p> </div>
Other	required literals	The corresponding character in the input string must exactly match the character.

* Control characters include all non-printing and formatting characters, such as line feeds and carriage returns.

Related reference

[Text Functions in Expressions \(on page 257\)](#)

Date Functions in Expressions

Date functions perform an operation on a date and time input value and return a string, numeric, or date and time value.

The following information describes the functions available for building expressions.



Tip:

- Any information within square brackets [] is optional.
- For more information on formatting strings, see **Conversion Functions**.
- If you want the current date/time associated with the UTC time zone, use the format `CurrentDateTime("UTC")`.

DATE

Description

Returns a date value representing a specified year, month, day, hour, minute, or second. The default value for date/time is always associated with the *current time zone*, unless a time zone is specified using a different function in the expression. Any actions performed in the same expression using this resultant value uses the current or specified time zone for evaluation.

**Note:**

If this function is used as the return value of an expression, the DateTime will be in universal time (UTC).

Format

DATE(int_year, int_month, int_day, [, int_hour, int_minute, int_second], [, timezone])

Parameters	Required?	Data Type	Description
int_year	Yes	Numeric	Integer for the year field of the date.
int_month	Yes	Numeric	Integer for the month field of the date.
int_day	Yes	Numeric	Integer for the day field of the date.
int_hour	No	Numeric	Integer for the hour field of the date.
int_minute	No	Numeric	Integer for the minute field of the date.
int_second	No	Numeric	Integer for the second field of the date.
timezone	No	String	Abbreviation of the time zone of the date.

Example

DATE(2008,10,31,11,59,59)

DATE(2008,10,31)

DATE(2008,10,31,11,59,59,"MST")

CurrentDateTime

Description

Returns the current date and time. The default value for date/time is always associated with the *current time zone*, unless a time zone is specified using a different function in the expression. Any actions performed in the same expression using this resultant value uses the current or specified time zone for evaluation.

To work with UTC values, you can use the UTC time zone. You can do this by specifying it in a DateTime function, or by converting the DateTime value to the UTC time zone. For example, the DateTime function can be expressed as CurrentDateTime("UTC") or as ConvertTimeZone({dateTimeVariable}, "UTC").



Note:

If this function is used as the return value of an expression, the DateTime will be in universal time (UTC).

Format

CurrentDateTime([timezone])

Parameters	Required?	Data Type	Description
timezone	No	String	Time zone to get the current date and time from.

DateTimeCompare

Description

Compares two Date-Time values and returns an indication of their relative values; that is, an integer that indicates whether the first instance is earlier than, the same as, or later than the second instance.

Format

DateTimeCompare(dateTime1, dateTime2)

Parameters	Required?	Data Type	Description
dateTime1	Yes	DateTime	First date and time value to compare.

Parameters	Required?	Data Type	Description
dateTime2	Yes	DateTime	Second date and time value to compare.

Example

```
DateTimeCompare(DATE(1899,10,31,11,59,59) , DATE(1999,10,31,11,59,59) ) < 0
```

```
DateTimeCompare(DATE(1999,10,31,11,59,59) , DATE(1999,10,31,11,59,59) ) = 0
```

```
DateTimeCompare(DATE(1999,10,31,11,59,59) , DATE(1899,10,31,11,59,59) ) > 0
```

Today**Description**

Returns the current date as a date and time value. The default value for date/time is always associated with the *current time zone*, unless a time zone is specified using a different function in the expression. Any actions performed in the same expression using this resultant value uses the current or specified time zone for evaluation. The time part of the day is zero (midnight).

**Note:**

If this function is used as the return value of an expression, the DateTime will be in universal time (UTC).

Format

```
TODAY([timezone])
```

Parameters	Required?	Data Type	Description
timezone	No	String	Time zone to get the current date from.

Year**Description**

Returns the year corresponding to the specified date.

**Tip:**

The DateTime input parameter takes into account the associated time zone of the input during evaluation.

Format

YEAR(date)

Parameters	Required?	Data Type	Description
date	Yes	DateTime	Date to get the year from.

Month**Description**

Returns a number between 1 and 12 (inclusive) representing the month of the year, corresponding to the specified date.

**Tip:**

The DateTime input parameter takes into account the associated time zone of the input during evaluation.

Format

MONTH(date)

Parameters	Required?	Data Type	Description
date	Yes	DateTime	Date to get the month from.

Day**Description**

Returns a number between 1 and 31 (inclusive) representing the day of the month, corresponding to the specified date.

Format

DAY(date)

Parameters	Required?	Data Type	Description
date	Yes	DateTime	Date to get the day from.

Hour

Description

Returns a number between 0 and 23 (inclusive) representing the hour of the day.

Example

Hour(CurrentDateTime("MST")) will give a result 2 off of Hour(CurrentDateTime("EST")).

Format

HOUR(date)

Parameters	Required?	Data Type	Description
date	Yes	DateTime	Date to get the hour from.

Minute

Description

Returns a number between 0 and 59 (inclusive) representing the minute of the hour.



Tip:

The DateTime input parameter takes into account the associated time zone of the input during evaluation.

Format

MINUTES(date)

Parameters	Required?	Data Type	Description
date	Yes	DateTime	Date to get the minute from.

Second

Description

Returns a number between 0 and 59 (inclusive) representing the second of the minute.

Format

SECOND(date)

Parameters	Required?	Data Type	Description
date	Yes	DateTime	Date to get the seconds from.

DatePart**Description**

Returns an integer value containing the specified component of a given DateTime value, such as the month or year.

Format

DatePart(component, date [, firstdayofweek [, firstweekofyear]])

Parameters	Required?	Data Type	Description
component	Yes	Numeric or String	Values representing which time component to extract: <ul style="list-style-type: none"> • 0 or "yyyy" = year • 1 or "q" = quarter • 2 or "m" = month • 3 or "y" = day of year (1 to 366) • 4 or "d" = day of month (1 to 31) • 5 or "ww" = week of year (1 to 53) • 6 or "w" = day of Week (1 to 7) • 7 or "h" = hour • 8 or "n" = minute • 9 or "s" = second
date	Yes	DateTime	Value to extract from.

Parameters	Required?	Data Type	Description
firstdayofweek	No	Numeric	<p>Values indicating which day to consider as the first day of the week:</p> <ul style="list-style-type: none"> • 0 = System setting • 1 = Sunday (default) • 2 = Monday • 3 = Tuesday • 4 = Wednesday • 5 = Thursday • 6 = Friday • 7 = Saturday
firstweekofyear	No	Numeric	<p>Values indicating whether to consider the first week of a year to be:</p> <ul style="list-style-type: none"> • 0 = System setting • 1 = week in which Jan 1 occurs • 2 = week that has at least 4 days in the new year • 3 = first full week in the new year

DateOnly

Description

Gets the date component of a DateTime value, with the time zeroed out.

**Tip:**

The DateTime input parameter takes into account the associated time zone of the input during evaluation.

Format

DateOnly(date)

Parameters	Required?	Data Type	Description
date	Yes	DateTime	Date and time from which the date is isolated.

ConvertTimeZone**Description**

Changes the time zone associated with a DateTime value. This function does not change the specific time being referenced; that is, the universal (UTC) time remains the same. A DateTime value that is associated with the time zone is returned.

To work with UTC values, you can use the UTC time zone. You can do this by specifying it in a DateTime function, or by converting the DateTime value to the UTC time zone. For example, the DateTime function can be expressed as `CurrentDateTime("UTC")` or as `ConvertTimeZone({dateTimeVariable}, "UTC")`.

Example

If 3:00 p.m. MST is converted to EST, then the result displayed is 5:00 p.m. EST. The time does not change, but it is associated with a different time zone.

Format

ConvertTimeZone(date, timeZoneID)

Parameters	Required?	Data Type	Description
date	Yes	DateTime	Date and time to associate with a timezone.
timeZoneID	Yes	String	Represents the time zone.

SpecifyTimeZone

Description

Specifies the time zone of a time value. A DateTime value that is associated with the local time is returned.

Example

If 3:00 p.m. MST is specified as EST, then the result is 3:00 p.m. EST. The time of the output value is the same as the input value.

Format

SpecifyTimeZone(date, timeZoneID)

Parameters	Required?	Data Type	Description
date	Yes	DateTime	Date and time to associate with a time zone.
timeZoneID	Yes	String	Represents the time zone.

GetTimeZone

Description

Gets the time zone the date is associated with. A string value that represents the associated time zone is returned.

Format

GetTimeZone(date)

Parameters	Required?	Data Type	Description
date	Yes	DateTime	Date and time to get the associated time zone from.

Related reference

[Conversion Functions in Expressions \(on page 255\)](#)

[Time Zone Abbreviations for Date and Time Functions \(on page 275\)](#)

[Time Span Functions in Expressions \(on page 281\)](#)

Time Zone Abbreviations for Date and Time Functions

Workflow recognizes that company operations may exist across time zones; therefore, you can build logic into your workflows that allows them to recognize multiple time zones and ensures that production is not affected.

If no time zone is specified when you develop your workflows, the time is set to the local computer that executes the workflow. The following table describes the accepted time zone abbreviations and their corresponding time zones.



Note:

Not all time zones are listed. If a time zone is not listed, you can use the full time zone ID that is displayed in the following registry:

- HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Time Zones

Abbreviation	Full Name	Registry ID	Description
ACDT	Australian Central Daylight Time	Cen. Australia Standard Time	(GMT+09:30) Adelaide
ACST	Australian Central Standard Time	AUS Central Standard Time	(GMT+09:30) Darwin
AEDT	Australian Eastern Daylight Time	AUS Eastern Standard Time	(GMT+10:00) Canberra, Melbourne, Sydney
AEST	Australian Eastern Standard Time	E. Australia Standard Time	(GMT+10:00) Brisbane
AFT	Afghanistan Time	Afghanistan Standard Time	(GMT+04:30) Kabul
AKST	Alaska Standard Time	Alaskan Standard Time	(GMT-09:00) Alaska
AMST	Armenia Standard Time	Armenian Standard Time	(GMT+04:00) Yerevan
ART	Argentina Standard Time	Argentina Standard Time	(GMT-03:00) Buenos Aires

Abbreviation	Full Name	Registry ID	Description
AST	Atlantic Standard Time	Atlantic Standard Time	(GMT-04:00) Atlantic Time (Canada)
ARAB	Arab Standard Time	Arab Standard Time	(GMT+03:00) Kuwait, Riyadh
ARABIA	Arabian Standard Time	Arabian Standard Time	(GMT+04:00) Abu Dhabi, Muscat
ARABIC	Arabic Standard Time	Arabic Standard Time	(GMT+03:00) Baghdad
AWST	Australian Western Standard Time	W. Australia Standard Time	(GMT+08:00) Perth
AZOST	Azores Standard Time	Azores Standard Time	(GMT-01:00) Azores
AZT	Azerbaijan Time	Azerbaijan Standard Time	(GMT+04:00) Baku
CAST	Central America Standard Time	Central America Standard Time	(GMT-06:00) Central America
CAT	Central Asia Time	Central Asia Standard Time	(GMT+06:00) Astana, Dhaka
CBST	Central Brazilian Time	Central Brazilian Standard Time	(GMT-04:00) Manaus
CET	Central Europe Time	Central Europe Standard Time	(GMT+01:00) Belgrade, Bratislava, Budapest, Ljubljana, Prague
CEST	Central European Standard Time	Central European Standard Time	(GMT+01:00) Sarajevo, Skopje, Warsaw, Zagreb
CCS	Caucasus Time	Caucasus Standard Time	(GMT+04:00) Caucasus Standard Time
CGT	Greenland Standard Time	Greenland Standard Time	(GMT-03:00) Greenland
CHN	China Standard Time	China Standard Time	(GMT+08:00) Beijing, Chongqing, Hong Kong, Urumqi

Abbreviation	Full Name	Registry ID	Description
CPST	Central Pacific Standard Time	Central Pacific Standard Time	(GMT+11:00) Magadan, Solomon Is., New Caledonia
CST	Central Standard Time	Central Standard Time	(GMT-06:00) Central Time (US & Canada)
CSTM	Central Standard Time (Mexico)	Central Standard Time (Mexico)	(GMT-06:00) Guadalajara, Mexico City, Monterrey - New
CVT	Cape Verde Time	Cape Verde Standard Time	(GMT-01:00) Cape Verde Is.
DST	Dateline Standard Time	Dateline Standard Time	(GMT-12:00) International Date Line West
EAT	East Africa Time	E. Africa Standard Time	(GMT+03:00) Nairobi
EET	Eastern European Time	E. Europe Standard Time	(GMT+02:00) Minsk
EGY	Egypt Standard Time	Egypt Standard Time	(GMT+02:00) Cairo
EKB	Ekaterinburg Standard Time	Ekaterinburg Standard Time	(GMT+05:00) Ekaterinburg
ESAST	South America Eastern Standard Time	E. South America Standard Time	(GMT-03:00) Brasilia
EST	Eastern Standard Time	Eastern Standard Time	(GMT-05:00) Eastern Time (US & Canada)
FJT	Fiji Standard Time	Fiji Standard Time	(GMT+12:00) Fiji, Kamchatka, Marshall Is.
FLE	FLE Standard Time	FLE Standard Time	(GMT+02:00) Helsinki, Kyiv, Riga, Sofia, Tallinn, Vilnius
GET	Georgia Standard Time	Georgian Standard Time	(GMT+03:00) Tbilisi
GMT	Greenwich Mean Time	GMT Standard Time	(GMT) Greenwich Mean Time : Dublin, Edinburgh, Lisbon, London

Abbreviation	Full Name	Registry ID	Description
GST	Greenwich Standard Time	Greenwich Standard Time	(GMT) Monrovia, Reykjavik
GTB	GTB Standard Time	GTB Standard Time	(GMT+02:00) Athens, Bucharest, Istanbul
HAST	Hawaii-Aleutian Standard Time	Hawaiian Standard Time	(GMT-10:00) Hawaii
IRST	Iran Standard Time	Iran Standard Time	(GMT+03:30) Tehran
IST	India Standard Time	India Standard Time	(GMT+05:30) Chennai, Kolkata, Mumbai, New Delhi
ISRAEL	Israel Standard Time	Israel Standard Time	(GMT+02:00) Jerusalem
JRD	Jordan Standard Time	Jordan Standard Time	(GMT+02:00) Amman
JST	Japan Standard Time	Tokyo Standard Time	(GMT+09:00) Osaka, Sapporo, Tokyo
KST	Korea Standard Time	Korea Standard Time	(GMT+09:00) Seoul
MAST	Mid-Atlantic Standard Time	Mid-Atlantic Standard Time	(GMT-02:00) Mid-Atlantic
MEST	Middle East Standard Time	Middle East Standard Time	(GMT+02:00) Beirut
MMT	Myanmar Standard Time	Myanmar Standard Time	(GMT+06:30) Yangon (Rangoon)
MRC	Morocco Standard Time	Morocco Standard Time	(GMT) Casablanca
MSK	Moscow Standard Time	Russian Standard Time	(GMT+03:00) Moscow, St. Petersburg, Volgograd
MST	Mountain Standard Time	Mountain Standard Time	(GMT-07:00) Mountain Time (US & Canada)
MSTM	Mountain Standard Time (Mexico)	Mountain Standard Time (Mexico)	(GMT-07:00) Chihuahua, La Paz, Mazatlan - New

Abbreviation	Full Name	Registry ID	Description
MUT	Mauritius Time	Mauritius Standard Time	(GMT+04:00) Port Louis
MVD	Montevideo Standard Time	Montevideo Standard Time	(GMT-03:00) Montevideo
NAEST	North Asia East Standard Time	North Asia East Standard Time	(GMT+08:00) Irkutsk, Ulaan Bataar
NAST	North Asia Standard Time	North Asia Standard Time	(GMT+07:00) Krasnoyarsk
NCAST	North Central Asia Standard Time	North Central Asia Standard Time	(GMT+06:00) Almaty, Novosibirsk
NLT	Newfoundland Time	Newfoundland Standard Time	(GMT-03:30) Newfoundland
NMT	Namibia Time	Namibia Standard Time	(GMT+02:00) Windhoek
NPT	Nepal Time	Nepal Standard Time	(GMT+05:45) Kathmandu
NZST	New Zealand Standard Time	New Zealand Standard Time	(GMT+12:00) Auckland, Wellington
PSA	Pacific South America Time	Pacific SA Standard Time	(GMT-04:00) Santiago
PST	Pacific Standard Time	Pacific Standard Time	(GMT-08:00) Pacific Time (US & Canada)
PSTM	Pacific Standard Time (Mexico)	Pacific Standard Time (Mexico)	(GMT-08:00) Tijuana, Baja California
PAK	Pakistan Standard Time	Pakistan Standard Time	(GMT+05:00) Islamabad, Karachi
ROM	Romance Standard Time	Romance Standard Time	(GMT+01:00) Brussels, Copenhagen, Madrid, Paris
SAEST	SA Eastern Standard Time	SA Eastern Standard Time	(GMT-03:00) Georgetown

Abbreviation	Full Name	Registry ID	Description
SAPST	SA Pacific Standard Time	SA Pacific Standard Time	(GMT-05:00) Bogota, Lima, Quito, Rio Branco
SAST	South Africa Standard Time	South Africa Standard Time	(GMT+02:00) Harare, Pretoria
SAWST	SA Western Standard Time	SA Western Standard Time	(GMT-04:00) La Paz
SEA	SE Asia Standard Time	SE Asia Standard Time	(GMT+07:00) Bangkok, Hanoi, Jakarta
SLST	Sri Lanka Standard Time	Sri Lanka Standard Time	(GMT+05:30) Sri Jayawardenepura
SMO	Samoa Time	Samoa Standard Time	(GMT-11:00) Midway Island, Samoa
SSK	Saskatchewan Time	Canada Central Standard Time	(GMT-06:00) Saskatchewan
SST	Singapore Standard Time	Singapore Standard Time	(GMT+08:00) Kuala Lumpur, Singapore
TON	Tonga Standard Time	Tonga Standard Time	(GMT+13:00) Nuku'alofa
TSM	Tasmania Standard Time	Tasmania Standard Time	(GMT+10:00) Hobart
TWT	Taiwan Time	Taipei Standard Time	(GMT+08:00) Taipei
USEST	US Eastern Standard Time	US Eastern Standard Time	(GMT-05:00) Indiana (East)
USMST	US Mountain Standard Time	US Mountain Standard Time	(GMT-07:00) Arizona
UTC	Coordinated Universal Time	N/A	(GMT)
VNZ	Venezuela Time	Venezuela Time Zone	(GMT-04:30) Caracas
VVS	Vladivostok Standard Time	Vladivostok Standard Time	(GMT+10:00) Vladivostok

Abbreviation	Full Name	Registry ID	Description
WAT	West Asia Standard Time	West Asia Standard Time	(GMT+05:00) Tashkent
WCAST	Western Central Africa Standard Time	W. Central Africa Standard Time	(GMT+01:00) West Central Africa
WET	Western European Time	W. Europe Standard Time	(GMT+01:00) Amsterdam, Berlin, Bern, Rome, Stockholm, Vienna
WPT	West Pacific Standard Time	West Pacific Standard Time	(GMT+10:00) Guam, Port Moresby
YAKT	Yakutsk Standard Time	Yakutsk Standard Time	(GMT+09:00) Yakutsk

Time Span Functions in Expressions

Time span values represent an interval of time. Time span functions perform operations on these values.

The following information describes the functions available for building expressions.



Tip:

- Any information within square brackets [] is optional.

TimeOfDay

Description

Gets the time components of a DateTime value. Only the time part of the date is associated with the time zone.

Format

TimeOfDay(date)

Parameters	Required?	Data Type	Description
date	Yes	DateTime	Date/time to isolate the time from.

Time

Description

Creates a time value based on time components. The return value is a time span value.

Format

Time(days, hours, minutes, seconds, milliseconds)

Parameters	Required?	Data Type	Description
days	Yes	Numeric	Number of days to include in the time value.
hours	Yes	Numeric	Number of hours to include in the time value.
minutes	Yes	Numeric	Number of minutes to include in the time value.
seconds	Yes	Numeric	Number of seconds to include in the time value.
milliseconds	Yes	Numeric	Number of milliseconds to include in the time value.

FromDays

Description

Creates a TimeSpan value by specifying the number of days in a time span.

Format

FromDays(days)

Parameters	Required?	Data Type	Description
days	Yes	Numeric	Number of days.

FromHours

Description

Creates a TimeSpan value by specifying the number of hours in a time span.

Format

FromHours(hours)

Parameters	Required?	Data Type	Description
hours	Yes	Numeric	Number of hours.

FromMinutes

Description

Creates a TimeSpan value by specifying the number of minutes in a time span.

Format

FromMinutes(minutes)

Parameters	Required?	Data Type	Description
minutes	Yes	Numeric	Number of minutes.

FromSeconds

Description

Creates a TimeSpan value by specifying the number of seconds in a time span.

Format

FromSeconds(seconds)

Parameters	Required?	Data Type	Description
seconds	Yes	Numeric	Number of seconds.

FromMilliseconds

Description

Creates a TimeSpan value by specifying the number of milliseconds in a time span.

Format

FromMilliseconds(milliseconds)

Parameters	Required?	Data Type	Description
milliseconds	Yes	Numeric	Number of milliseconds.

Days

Description

Returns the days component of a time span value.

Format

Days(timespan)

Parameters	Required?	Data Type	Description
timespan	Yes	TimeSpan	Time to extract the component from.

Hours

Description

Returns the hours component of a time span value.

Format

Hours(timespan)

Parameters	Required?	Data Type	Description
timespan	Yes	TimeSpan	Time to extract the component from.

Minutes

Description

Returns the minutes component of a time span value.

Format

Minutes(timespan)

Parameters	Required?	Data Type	Description
timespan	Yes	TimeSpan	Time to extract the component from.

Seconds

Description

Returns the seconds component of a time span value.

Format

Seconds(timespan)

Parameters	Required?	Data Type	Description
timespan	Yes	TimeSpan	Time to extract the component from.

Milliseconds

Description

Returns the milliseconds component of a time span value.

Format

Milliseconds(timespan)

Parameters	Required?	Data Type	Description
timespan	Yes	TimeSpan	Time to extract the component from.

TotalDays

Description

Converts a TimeSpan value to a number of days. The return type is a double.

Format

TotalDays(timespan)

Parameters	Required?	Data Type	Description
timespan	Yes	TimeSpan	Total time span value expressed as a number of days.

TotalHours

Description

Converts a TimeSpan value to a number of hours. The return type is a double.

Format

TotalHours(timespan)

Parameters	Required?	Data Type	Description
timespan	Yes	TimeSpan	Total time span value expressed as a number of hours.

TotalMinutes

Description

Converts a TimeSpan value to a number of minutes. The return type is a double.

Format

TotalMinutes(timespan)

Parameters	Required?	Data Type	Description
timespan	Yes	TimeSpan	Total time span value expressed as a number of minutes.

TotalSeconds

Description

Converts a TimeSpan value to a number of seconds. The return type is a double.

Format

TotalSeconds(timespan)

Parameters	Required?	Data Type	Description
timespan	Yes	TimeSpan	Total time span value expressed as a number of seconds.

TotalMilliseconds

Description

Converts a TimeSpan value to a number of milliseconds. The return type is a double.

Format

TotalMilliseconds(timespan)

Parameters	Required?	Data Type	Description
timespan	Yes	TimeSpan	Total time span value expressed as a number of milliseconds.

Duration

Description

Returns the absolute value of a time span.

Format

Duration(timespan)

Example

Time(0,-5,0,0,0) represents -5 hours. Using Duration, you get a time span that represents +5 hours:

- Duration(Time(0,-5,0,0,0)) = Time(0,5,0,0,0)
- Duration(FromHours(-5)) = FromHours(5)

Parameters	Required?	Data Type	Description
timespan	Yes	TimeSpan	Time span value to get the absolute value of.

Expression Operators

An operator performs an operation in an expression on one or more elements that hold value.

The following table describes the operators that can be used to build your expressions:

Symbol	Description	Example	Supported Date Types (non-numeric & non-boolean)
+	add/concatenate	A + B	string+string=string date+time=date

Symbol	Description	Example	Supported Date Types (non-numeric & non-boolean)
			time+time=time
-	subtract/negate	A - B or -A	date-date=time date-date=date time-time=time -time=time
*	multiply	A * B	N/A
/	divide	A / B	N/A
==	is equal to*	A == B	string==string date==date time==time
!=	is not equal to*	A != B	string!=string date!=date time!=time
<	less than*	A < B	string<string date<date time<time
<=	less than or equal to*	A <= B	string<=string date<=date time<=time
>	greater than*	A > B	string>string date>date time>time

Symbol	Description	Example	Supported Date Types (non-numeric & non-boolean)
>=	greater than or equal to*	A >= B	string>=string date>=date time>=time
!	logical NOT	A ! B	N/A
&&	logical AND	A && B	N/A
	logical OR	A B	N/A
	bitwise inclusive OR	A B	N/A
!&	bitwise exclusive OR	A !& B	N/A
&	bitwise AND	A & B	N/A
^	power (exponent)	A ^ B	N/A
%	modulus (remainder)	A % B	N/A

* Comparison operators are not case sensitive. For case sensitive comparisons, use the function

`IsStringEqual`.

Expression Variables

Variables are previously configured components in Workflow models that can be included in an expression for evaluation.

The variables available in the Expression Editor are dependent on prior configurations.

For more information, see:

- **Condition Events Expressions**
- **Workflow Expressions**

Variables and Dates

The default value for date/time is always associated with the *current time zone*, unless a time zone is specified using a different function in the expression. Any actions performed in the same expression using this resultant value uses the current or specified time zone for evaluation.

String, Numeric, and Boolean Literals in Expressions

A literal is a textual representation of a particular value in an expression.

In Workflow, a literals can be Numeric, String, or Boolean.

String Literals

String literals consists of zero or more characters surrounded by quotation marks. If a string contains quotation marks, these must be escaped in order for the expression to parse.

Escape Sequence

\ " Resolves to a quotation mark.

\\ Resolves to a backslash. Any other character following the \ results in an unrecognized escaped character error.



Important:

Projects created in a version earlier than v1.2 that use \ in a string literal in an expression must be changed to \\.

Numeric Literals

Numeric literals represent numbers, including positive and negative integers (for example, 1024) and decimal numbers (for example, 1.9867543).

These literals can also be defined by using scientific notation:

- [-] m. ddddd E+ xx
- [-] m. ddddd E- xx
- [-] m. ddddd e+ xx
- [-] m. ddddd e- xx

One or more nonzero digits (m) precede the decimal separator ("."). A minus sign ("-") can precede m. The Type performing the conversion determines the number of decimal places (dddddd) in the string, and

maximum and minimum values for xx and m . The exponent ($+ / - xx$) consists of either a plus or minus sign followed by at least one digit.

Boolean Literals

Boolean literals represent logical value, and can be either True or False. The expression language is not case-sensitive and any combination of upper- and lowercase letters are valid.

Chapter 5. Configure Your Environment

User Password Configuration

Workflow User Password Configuration

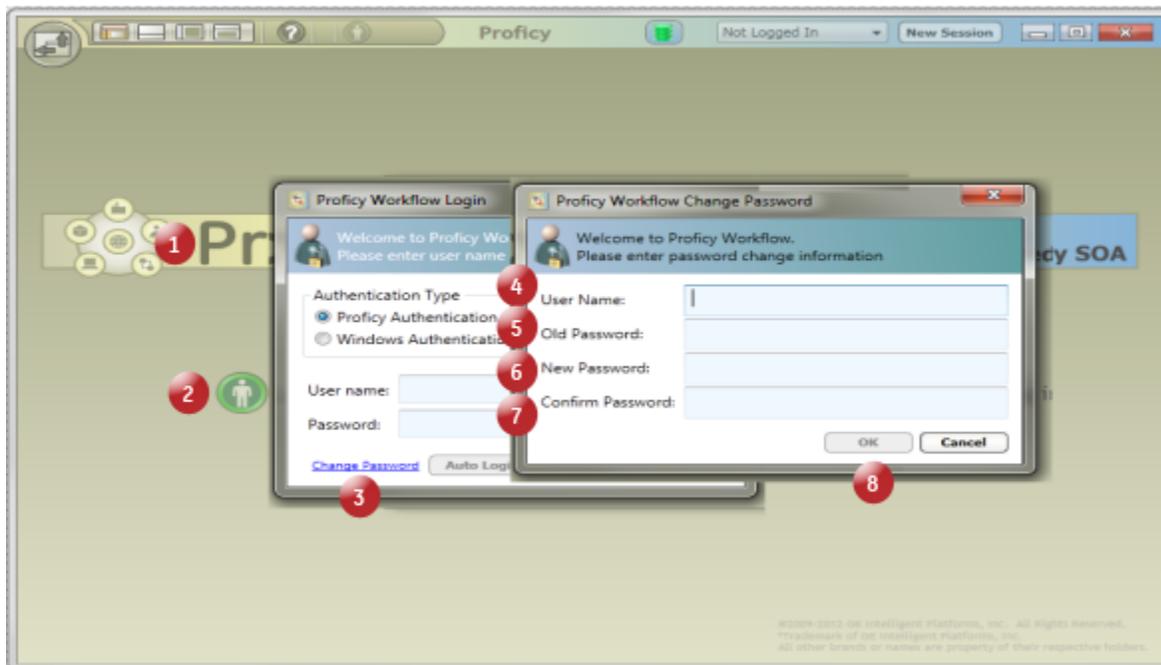
Workflow provides the ability for individual users to change their login password at any time. This ability applies only to Workflow users; it does not apply to Windows users.

Change Workflow user password

You can change the password for your Workflow user account at any time. This ability does not apply to Windows user accounts.

About this task

Your Administrator must enable the change password capability in order for you to change your password.



Procedure

1. Launch Workflow or, if you are already logged in, log out.
2. On the main Workflow screen, click **Log in to Workflow**.
The **Workflow Login** dialog box appears.

3. Click **Change Password**.

The **Proficy Workflow Change Password** dialog box appears.

4. In the **User Name** field, enter the user name you use to log in to Workflow.
5. In the **Old Password** field, enter your current password.
6. In the **New Password** field, enter a new, unique password.
7. In the **Confirm Password** field, re-enter the new password.

If the password does not match what you typed in the **New Password** field, the **OK** button is not enabled.

8. Click **OK**.

New Password Configuration

The **Change Password** dialog box is used by individual Workflow users who want to change their password. It is accessed from the login dialog box, but is available only if the Administrator has enabled the feature.

The following table describes the information available for changing your password.

Attribute	Value	Description
User Name	User-defined	Specifies the unique login name for the current user.

Attribute	Value	Description
Old Password	User-defined	Specifies the current password for this user.
New Password	User-defined	Specifies a new unique password for the current user.
Confirm Password	User-defined	Specifies the same unique password that you entered in the New Password field.

Computer Management

Computer Management

Workflow access and layout features allow you to configure lockdown state, default startup options, and user overrides for individual computers, all computers in your system, or a subset of the computers in your system.

Lockdown mode and equipment linking are not available for global default configurations.

You can configure access and layouts for all users of a workstation and you can override certain features for one or more specific users.

Global computer configurations allow you to customize, from a single location, the way Workflow opens and appears on all computers in your system or on a subset of computers. That is, you are not required to set the computer layout and startup display for each individual computer.

Overrides for individual users can be defined on a global basis; that is, if you configure an override for a specific user, that user can log on to any computer in the system that is part of the global configuration and see the layout and startup display configured for him or her.

Global computer settings take precedence over individual computer settings for all computers that are included in a global default configuration. If you do not want a specific computer to use the default configuration, you can either exclude it during the initial default set up or you can override the default configuration.

Access and Layout Configurations

Depending on which computer editor you access, Workflow will provide an overview of the access and layout configurations for individual computers in your system, or will allow you to configure all of the computers or a subset of computers in your system to use global startup options.

If a computer is included in a global default configuration, any settings you configured specifically for that computer are overridden by the global default settings. If you do not want a specific computer to use the default configuration, you can either exclude it during the initial default set up or you can override the default configuration.

Use or Override the Global Default Configuration

You can choose to include or exclude an individual computer from a global default configuration.

Procedure

1. In the navigator, click **Proficiency System > Proficiency System > Computers**.
2. From the **Computers** list, select the computer that you want to configure.



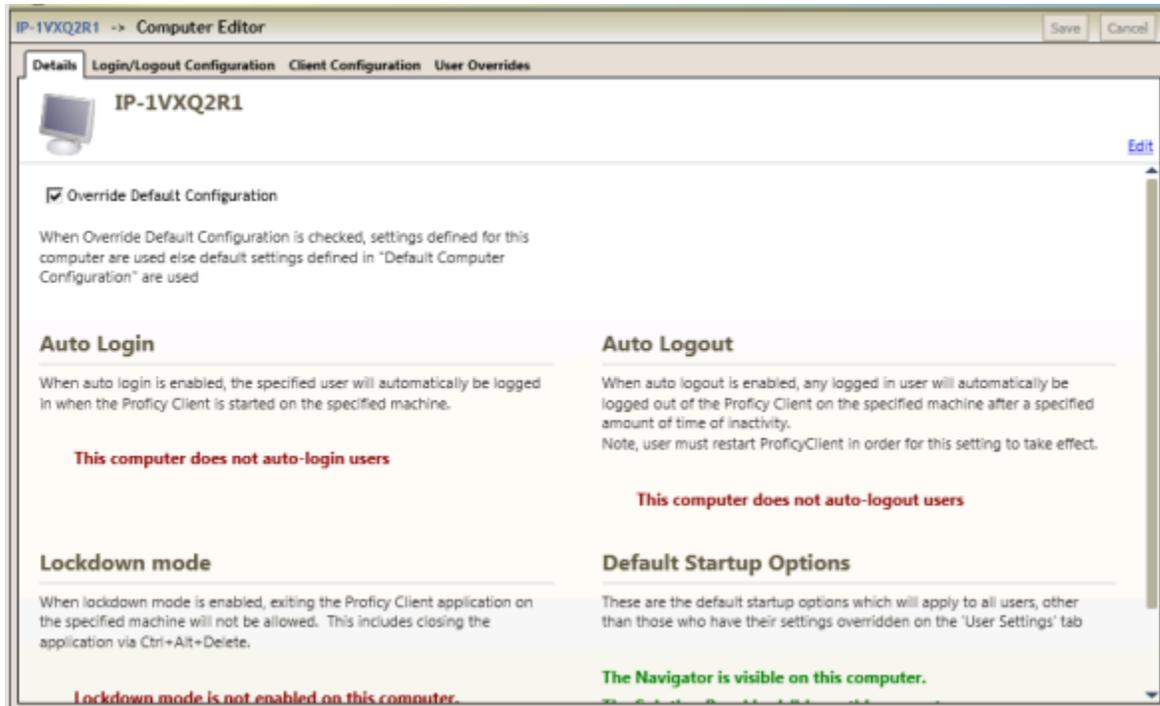
Tip:

If you are configuring settings for all computers in your system or a subset of those computers, you can skip this step. However, you can still configure global default settings if you select an individual computer.

3. In the **Displays** panel, select the **Computer Editor**.
4. Click the **Details** tab.
5. Select the **Override Default Configuration** check box to allow the selected computer to use startup options defined specifically for it; otherwise, clear the check box to use the global default configuration settings.
6. Click **Save**.

Computer Configuration Summary

The **Details** tab of the **Computer Editor** provides summary information about the selected computer's configuration.



Edit Description

You can change the description of this resource.

Attribute	Value	Description
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

Details Tab

The following table describes the available options.

Attribute	Value	Description
Override Default Configuration	N/A	Select this check box to indicate that the settings for the selected computer will override the global default configuration and use startup options configured specifically for it. Clear this check box to indicate that the selected

Attribute	Value	Description
		computer will use the global default configuration.
Auto Login	N/A	Displays whether the computer is configured for automatic login for any users.
Auto Logout	N/A	Displays whether the computer is configured for automatic logout for any users.
Lockdown mode	N/A	<p>Displays whether the lockdown mode is configured on the computer.</p> <p>When lockdown is enabled, you cannot access anything outside of the Workflow client, but may log in or log out. When lockdown is disabled, you can access other programs, minimize the client, and so on.</p>
Default Startup Options	N/A	Displays the startup options that are configured on the computer.

Select Computers for Global Default Configuration

From a single location, you can choose the computers in your system that will use the global default configuration.

Procedure

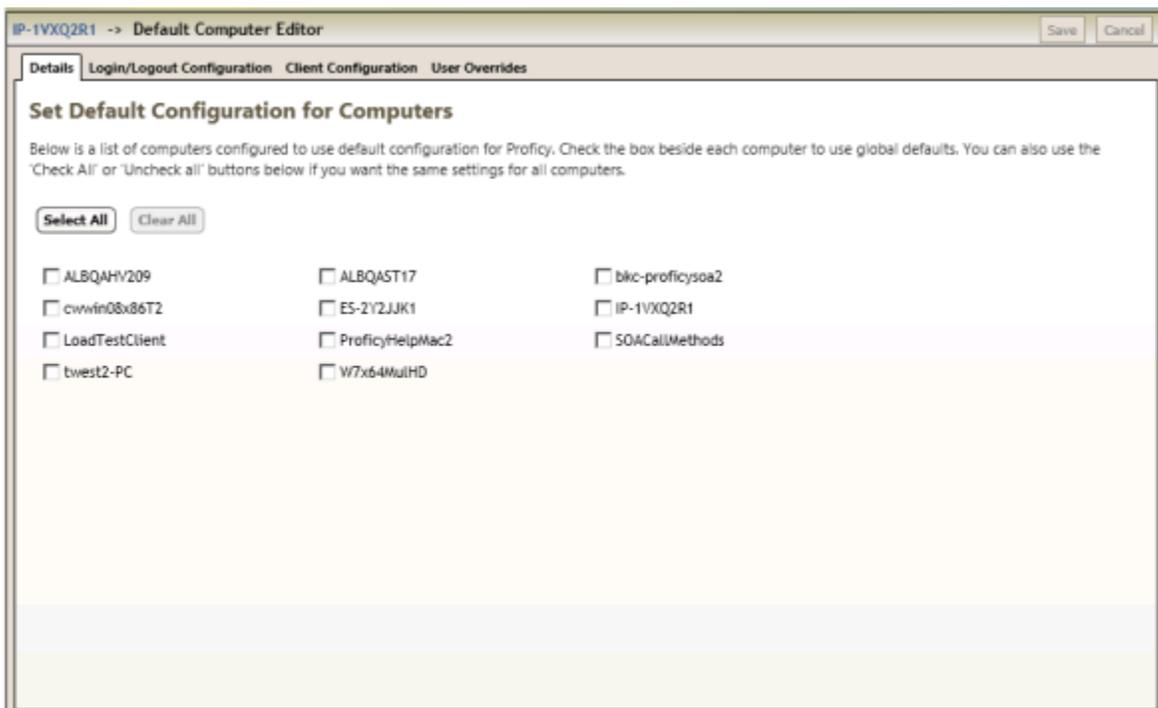
1. In the navigator, click **Proficy System > Proficy System > Computers**.
2. In the **Displays** panel, select the **Default Computer Editor**.
3. Click the **Details** tab.
4. Do one of the following:

- To configure all computers in your system to use the default configuration, click **Select All**.
- To configure individual computers to use the default startup display, select the check box for the specific computer(s).
- To stop using the default configuration, click **Clear All**.

5. Click **Save**.

Global Computer Configuration

The **Details** tab of the **Default Computer Editor** provides the ability to select the computers that will use the global default computer settings.



The following table describes the available options.

Attribute	Value	Description
Select All	N/A	Click to select all of the computers in your system to use the global default configuration. Clear individual check boxes to indicate that the corresponding computer will not use the global default configuration.

Attribute	Value	Description
Clear All	N/A	Click to clear all of the check boxes. Select individual check boxes to indicate that the corresponding computers will use the global default configuration.
List of Computers	System-defined	Displays the names of all of the computers in your system.

Startup Options

The client layout configuration allows you to customize the way Workflow opens and appears on a specific computer, on all computers in your system, or on a subset of those computers.

Client layouts, lockout mode, and equipment associations are configured for all users that log on to individual workstations. Layouts, startup objects, and startup displays and solutions are configured for all users that log on to any computer that is included in a global default configuration.

Lockdown Mode

When lockdown is enabled, you cannot access anything outside of the Workflow client, but may log in or log out. When lockdown is disabled, you can access other programs, minimize the client, and so on.

Lockdown mode is not available when you use global default configurations.



Note:

- If any workstation keyboards in your facility have multimedia keys—for example, Internet or e-mail launch—these keyboard buttons may disable certain key macros or allow users to circumvent Workflow security measures. It is recommended that you reprogram or disable the software that operates those buttons. Refer to your computer's documentation for instructions on disabling these buttons.
- If any computers in your facility have lockdown mode enabled, you must ensure that web displays that are available to your users do not contain any links. This will help to prevent circumvention of your Workflow security measures.
- Because the online help contains external links, you cannot open it when the computer is in lockdown mode.

Equipment Link

If there is a piece of equipment associated with the selected computer workstation, you can create a link to that equipment resource.

Equipment linking is not available when you use global default configurations.

Default Startup Options

The default setup options allow you to specify the startup object and display that will be shown when Workflow is started on a specific workstation, or on all workstations included in a default configuration. These settings apply to all users except those who have the user overrides settings configured.

When creating the layout configuration for a workstation, including any user overrides, the following rules apply:

- You can select the navigator and/or object views for the user, as well as having specific navigation, startup object, and startup displays/solutions specified.
- If the navigator is not selected, you must specify a startup object, and if the solution panel is not selected, you must specify startup solution and display objects; otherwise, the configuration is considered invalid and an error is reported.
- If you specify a startup solution, you must also specify a startup display.

If you do not want a specific computer to use the default configuration, you can either exclude it during the initial default set up or you can override the default configuration.

Enable Lockdown Mode

Configure lockdown mode if you want to prevent users that are logged onto the selected computer from accessing any applications other than Workflow.

About this task

Use this procedure to prevent access applications other than Workflow.

Procedure

1. In the navigator, click **Proficy System > Proficy System > Computers**.
2. From the **Computers** list, select the computer that you want to configure.

**Tip:**

If you are configuring settings for all computers in your system or a subset of those computers, you can skip this step. However, you can still configure global default settings if you select an individual computer.

3. In the **Displays** panel, select the **Computer Editor**.
4. Click the **Client Configuration** tab.
5. In the Lockdown mode section, select the **Enable Lockdown mode** check box.
6. Click **Save**.

Associate an Equipment Resource with a Computer

Configure an equipment link if you want the selected computer to be associated with an equipment resource.

Procedure

1. In the navigator, click **Proficy System > Proficy System > Computers**.
2. From the **Computers** list, select the computer that you want to configure.

**Tip:**

If you are configuring settings for all computers in your system or a subset of those computers, you can skip this step. However, you can still configure global default settings if you select an individual computer.

3. In the **Displays** panel, select the **Computer Editor**.
4. Click the **Client Configuration** tab.
5. In the **Equipment Link** section, click **Link to Equipment**.
The **Universal Browser** appears.
6. Select the equipment resource you want to link to, and then click **OK**.

**Note:**

To remove the equipment link, click **Clear Equipment Link**.

7. Click **Save**.

8. Select the equipment resource you want to link to, and then click **OK**.

**Tip:**

To remove the equipment link, click **Clear Equipment Link**.

Configure Computer Startup Options

Configure the startup options to determine how Workflow will appear to users who log on to the selected computer.

Procedure

1. In the navigator, click **Proficiency System > Proficiency System > Computers**.
2. From the **Computers** list, select the computer that you want to configure.

**Tip:**

If you are configuring settings for all computers in your system or a subset of those computers, you can skip this step. However, you can still configure global default settings if you select an individual computer.

3. In the **Displays** panel, select the editor you want to use.

To configure startup options for...	Select
an individual computer	Computer Editor
all computers or a subset of computers in your system	Default Computer Editor

4. Click the **Client Configuration** tab.
5. **Optional:** Select or clear the **Show Solution Panel** check box and/or the **Show Navigator** check box.

**Note:**

If you clear these check boxes, you must define a startup solution and display.

6. **Optional:** To configure the startup object:
 - a. Select the **Navigate Users on Login** check box.
 - b. Click **Change Startup Object**, and then select the object to use as the startup object.
The name and path of the startup object appear.

7. **Optional:** To configure the startup solution and display:

- a. Select the **Users will be started in a specific solution/display** check box.
- b. Click **Change Startup Solution**, and then select the solution and display to use when the Workflow starts up.
The names of the startup solution and display appear.

8. Click **Save**.

Client Configuration

The **Client Configuration** tab on the **Computer Editor** allows you to enable or disable lockdown mode, to define the default startup options for a specific workstation, and to link a piece of equipment to a specific workstation.

On the **Default Computer Editor**, the **Client Configuration** tab allows you to define the default startup options for all of the computers in your system or a subset of those computers.



Note:

This tab is available in the Computer Editor only if the selected computer is configured to override the global configuration settings.

An information message is posted in the Feedback Zone any time a change is detected in the configuration settings of the functions.

Lockdown Mode

The following table describes the available options.

Attribute	Value	Description
Enable Lockdown mode	User-defined	<p>Select this check box to keep Workflow displayed at all times.</p> <p>When lockdown is enabled, you cannot access anything outside of the Workflow client, but may log in or log out. When lockdown is disabled, you can access oth-</p>

Attribute	Value	Description
		er programs, minimize the client, and so on.

Equipment Link

The following table describes the available options.

Attribute	Value	Description
Link to Equipment	User-defined	Click to open the Universal Browser to select an equipment location to associate with this computer.
Clear Equipment Link	N/A	Click to clear the equipment location association from the selected computer.

Default Startup Options

The following table describes the available options.

Attribute	Value	Description
Layout Configuration: Show Solution Pane	User-defined	Select this check box to specify whether the Solution Panel is displayed on the selected workstation when Workflow starts up.
Layout Configuration: Show navigator	User-defined	Select this check box to specify whether the navigator is displayed on the selected workstation when Workflow starts up.
Startup Object: Navigate Users on Login	User-defined	Select this check box to enable the Change Startup Object link.
Startup Object: Change Startup Object	User-defined	Click to open the Universal Browser to select the object that will be displayed on the select-

Attribute	Value	Description
		ed workstation when Workflow starts up.
Default Startup Options: Users will be started in a specific solution/display	User-defined	Select this check box to enable the Change Startup Solution link.
Default Startup Options: Change Startup Solution	User-defined	Click to open the Universal Browser to select the solution and display that will be displayed on the selected workstation when Workflow starts up.

Computer Customization

User overrides allow you to customize the way that the Workflow client opens and appears on a specific computer for a specific user. Client layouts, startup objects, and startup displays can be customized for individual users that log on to a specific workstation. Global user overrides allow you to customize the way that the Workflow client opens and appears on any computer that the selected user logs into, regardless of whether the computer is included in a default configuration.

Overrides that are configured using the Computer Editor are specific to the selected computer. Overrides that are configured using the Default Computer Editor are global; that is, the specified user can log on to any computer that is included in a global default configuration and the startup options defined for that user will be displayed.

You can specify the startup object in the navigator, as well as the solution and display that are shown at startup, regardless of the user context saved during the previous user session.

User overrides replace any default settings when the specific user logs onto this workstation. User overrides replace any default settings when the specific user logs onto any computer using any computer in the system, regardless of whether it is included in a default configuration.

When creating the layout configuration for a workstation, including any user overrides, the following rules apply:

- You can select the navigator and/or object views for the user, as well as having specific navigation, startup object, and startup displays/solutions specified.
- If the navigator is not selected, you must specify a startup object, and if the solution panel is not selected, you must specify startup solution and display objects; otherwise, the configuration is considered invalid and an error is reported.
- If you specify a startup solution, you must also specify a startup display.

If you do not want a specific computer to use the default configuration, you can either exclude it during the initial default set up or you can override the default configuration.

During startup, after the user has been authenticated, the user's context is loaded, based on the following rules:

- If a layout is configured for a particular user; that is, user overrides are configured, that layout is used to determine the context that is loaded. Otherwise, the default layout context for a specific computer is used. If global default configurations are used, overrides defined for a specific user take precedence.
- If a startup object has been specified, as well as the startup solution and display, then no user context from the previous session is loaded. The context is defined by the configured layout settings.
- Workflow tries to load the context from the previous session on this workstation, and then checks if the startup object or solution and display were specified in the layout. If any of these options were specified, then they are set as part of the user context to be restored.
- If there is no context associated with a particular workstation, then the last saved user context is loaded and checked against the layout settings.
- When a user logs in for the first time, no context is loaded. However, the client still checks the layout configuration to set the proper layout, startup navigation, solution, and display objects.

Layout Buttons

The layout buttons are located on the toolbar, and are available according to the layout configured for the specific workstation or according to the user overrides for the individual who is logged in on that workstation.



Note:

A layout for a specific workstation can also be based on default settings if the workstation is included in a global default configuration.

If you are working in a layout that exposes the Displays panel, you can maximize your working area by clicking the toggle arrow at the top of the panel, which alternately collapses or expands the panel with each click. In the collapsed mode, the panel appears as a thin vertical bar. Clicking on the bar causes the Displays panel to open in pop-up mode, from which you can select a different display. After selecting a new display, moving your cursor off the pop-up panel collapses it again.



Note:

The pop-up version of the panel does not allow you to choose between compatible or all displays. If you need to do so, you must click the toggle arrow to expand the full panel.

Override a Computer's Startup Options

You can override the configurations for a specific computer with specialized settings for individual users who log into this computer, or for computers with specialized settings for individual users who log into any computer configured to use global default settings.

Procedure

1. In the navigator, click **Proficy System > Proficy System > Computers**.
2. From the **Computers** list, select the computer that you want to configure.



Tip:

If you are configuring settings for all computers in your system or a subset of those computers, you can skip this step. However, you can still configure global default settings if you select an individual computer.

3. In the **Displays** panel, select the editor you want to use.

To configure user overrides for...	Select
an individual computer	Computer Editor
all computers or a subset of computers in your system	Default Computer Editor

4. Click the **User Overrides** tab.
5. In the **User Overrides** list, add a user you want to create overrides for.
6. **Optional:** Select or clear the **Show Solution Panel** check box and/or the **Show Navigator** check box.

**Note:**

If you clear these check boxes, you must define a startup solution and display.

7. **Optional:** To change the startup object:

- a. Select the **Navigate Users on Login** check box.
- b. Click **Change Startup Object**, and then select the object to use as the startup object.

The name and path of the startup object appear.

8. **Optional:** To change the startup solution and display:

- a. Select the **Users will be started in a specific solution/display** check box.
- b. Click **Change Startup Solution**, and then select the solution and display to use when the Workflow starts up.

The names of the startup solution and display appear.

9. Click **Save**.

Override Startup Options

The **User Overrides** tab allows you to override the startup options for specific users who log on to the selected computer or for specific users who log on to any computer that is included in a global default configuration.

**Note:**

This tab is available in the Computer Editor only if the selected computer is configured to override the global configuration settings.

Overrides that are configured using the **Computer Editor** are specific to the selected computer. Overrides that are configured using the **Default Computer Editor** are global; that is, the specified user can log on to any computer that is included in a global default configuration and the startup options defined for that user will be displayed.

The following tables describe the information available on the UserOverrides tab, where you can configure a custom client layout for individual users.

Attribute	Value	Description
List	User-defined	Displays a list of users that have user overrides configured on the selected computer.
Add User	N/A	Click to add a user that you can configure user overrides for on the selected computer.
Delete User	N/A	Click to delete a user from the overrides list. <div data-bbox="1024 688 1421 1144" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: When a user is removed from the User Overrides list, the default layout configured for the selected computer will be displayed the next time this user logs on to that computer. </div>

Layout Configuration

These options specify the client panels that will be visible on this computer when the selected user logs on.

Options	Description
Show Solution Panel	<p>Select this check box to show the Solution Panel when the selected user logs in on this computer.</p> <p>If you clear this check box, the Solution Panel is hidden for the selected user at log in on this computer, and you must select a startup solution and display for the user.</p>

Options	Description	
Show Navigator		<p>Select this check box to show the navigator when the selected user logs in on this computer.</p> <p>If you clear this check box, the navigator is hidden for the selected user at log in on this computer, and you must select a startup object for the user.</p>

Startup Object

These options specify the object that this computer will start at when the selected user logs on.

Options	Description	
Navigate Users on Login		<p>Select this check box to start the user at a specific object on this computer.</p> <div data-bbox="1026 1073 1419 1383" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: You must select this check box and define a startup object if the Show Solution Panel check box is cleared.</p> </div>
Change Startup Object		<p>Click to open the Universal Browser to select the startup object to display at log in for this user on this computer.</p> <div data-bbox="1026 1612 1419 1881" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: You must select a startup object if the Show Solution Panel check box is cleared.</p> </div>

Startup Display/Solution

These options specify the display and solution that will be displayed on this computer when the selected user logs on.

Options	Description	
Users will be started in a solution/display		<p>Select this check box to start the user with a specific display and solution on this computer.</p> <div data-bbox="1024 621 1419 932" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: You must select this check box and a start-up display/solution if the Show navigator check box is cleared.</p> </div>
Change Startup Solution		<p>Click to open the Universal Browser, where you can select the startup display/solution to display at login for this user on this computer.</p> <div data-bbox="1024 1203 1419 1474" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: You must select a start-up display/solution if the Show Navigator check box is cleared.</p> </div>

User Interface Controls in the Task List

User Interface Controls in the Task List

My Task List Editor allows you to select user controls that you want to be visible in the Task List, and copy these configurations to other computers or personnel classes.

Controls



Note:

This functionality is not applicable to the Mobile-sized Task List.

All controls of the Task List are visible unless configured otherwise. If the check box next to a control is cleared, the control will not be available during run time.



Tip:

All changes to the My Task List Editor section must be saved before you can copy the configuration to other computers or personnel classes.

The following table describes the available options.

Controls	Value	Description
Task Title	User-defined	Displays the given name for a workflow task.
Task Grid	User-defined	Provides an overview of tasks, including title, duration, expiry, and priority.
Task Context Navigator	User-defined	Provides the ability in real time to view and navigate all tasks currently running.
Task Step Listing	User-defined	Provides a visual display of the task steps being executed or that require execution.
Task Step Context Navigator	User-defined	Provides the ability in real time to view and navigate all tasks steps currently running.
Status Message	User-defined	Allows for display of a message that provides further information to the operator regarding the currently running task step.

Controls	Value	Description
Release Task Steps Dialog	User-defined	Allows unauthorized users to release task steps during run time. Clear this check box to prevent task step release by unauthorized users.
Task/Task History	User-defined	Displays the Tasks and Task History tabs in the control bar. Clear this check box to prevent navigation from current to past tasks during run time.
Task List Context	User-defined	Shows all information related to computer, user, and equipment context. Clearing this check box hides this information along with the Context button, and results in display of the preconfigured context for the task.
Equipment Context Editing	User-defined	Shows the Context button. Clear this check box to hide the button and allow operators to only view equipment context information during run time, not modify it.
Views	User-defined	Allows for interaction with manual inputs during run time, such as filling in a form, performing a manual start for a task, signing a document, or viewing a document. If no manual input for a workflow is required, then this button is not required.
Details	User-defined	Provides for viewing general read-only information, such as start/end times and person-

Controls	Value	Description
		nel/location assignments, about a selected task or task step.
Actions	User-defined	Provides for performing advanced actions such as cancelling, acquiring, and reassigning a task; entering comments; and jumping to other task steps. If this button is not available, these operations cannot be performed.
Filters	User-defined	Allows for task filtering by specific criteria such as task name, priority, state, assignment, and expiration.
Start Task	User-defined	Provides for starting a schedule that is associated with a task or task step during run time.
Step Layout	User-defined	Allows for configuring the view of the Task List during run time; that is, either horizontal or vertical.
Font Size	User-defined	Provides for changing the size of the text font during run time.

Copy to Other Resources

Click to open the **Universal Browser** to select one or more computers, users, or personnel classes to copy the selected visibility options for the My Task List Editor to.

Preview

Displays the layout of the My Task List Editor based on the visibility selections. You can view the selections in either the Horizontal Layout Mode or the Vertical Layout Mode.

Configure the Visible User Interface of the Task List

You can configure the visible user interface controls on the Task List through personnel class and computer resources.

Before you begin



Note:

If you are configuring the user interface for the Web Task List, you must use personnel classes.

Procedure

1. In the navigator, select one of the following paths, and then select the resource that you want to configure:
 - **Personnel > Classes**
 - **Proficiency System > Proficiency System > Computers**
2. In the **Displays** panel, select **My Task List Editor**.
3. In the **Visible** column, clear the check boxes of the controls that you do not want an operator to see in the Task List during run time.



Tip:

The appearance of the Task List displays in the **Preview** section. You can view the layout in either the Horizontal Layout Mode or the Vertical Layout Mode.

4. Click **Save**.

Copy Task List Layout Settings to Other Computers or Personnel Classes

You can copy the Task List user interface layout by using preconfigured personnel class and computer resources.

Procedure

1. In the navigator, select one of the following paths, and then select the resource whose settings you want to copy:
 - **Personnel > Classes**
 - **Proficiency System > Proficiency System > Computers**
2. In the **Displays** panel, select **My Task List Editor**.
3. **Optional:** In the **Visible** column, clear the check boxes of the elements that you do not want an operator to see in the Task List during run time, and then click **Save**.
4. In the **Copy to Other Resources** section, click the **Copy to** button.
The **Universal Browser** appears.

5. You can copy computer and/or personnel class settings, depending on the type of Task List your operators are using.

For the Task List:



Note:

If you have both computer and personnel classes configured, the system will default to the layout configured for personnel classes. You can configure either computer or personnel classes.

- a. In the navigator, click one or both of the following:
 - **Proficiency System > Computers**, and then select the computer that you want to copy resources to.
 - **Personnel > Classes**, and then select the class that you want to copy resources to.

For the Web Task List:



Note:

If you are copying settings for the Web Task List, you must use personnel classes.

- a. In the navigator, click **Personnel > Classes**.
 - b. In the **Personnel Classes** list, select the class that you want to copy resources to.
6. Click **OK**.

Custom Solutions and Displays

Custom User Solutions and Displays

Workflow provides the ability to create custom user solutions. User solutions allow you to organize the displays that you create.

User solutions are displayed in the solution panel as separate and distinct solutions. When you create a user solution, it automatically appears in the solution panel.

You can configure a user display to be:

- Available only in the navigator.
- Visible in the navigator and in the **Displays** panel under a user solution.
- Context aware, and if so, which resource and user solution it works with.

**Note:**

When a context aware resource is selected in the navigator, its compatible views are highlighted in the **Displays** panel. You can then select a compatible view or display to begin working with the resource. For more information, see **Display Panel**.

When a user solution is created, the **Display Editor** appears with a series of icons that can be selected to organize user displays. The selected icon appears beside the user solution name in the solution panel.

Configure Context Awareness for a User Display

You can configure context awareness by determining which resource and user solution that a display works with.

Before you begin

**Note:**

User displays and user solutions must have been added before you can associate them and configure context awareness. For more information, see:

- Add a Global Displays resource
- Add a user solution

Procedure

1. In the navigator, click **Global Displays > Displays**.
2. From the next list in the hierarchy, select the resource you want to work with.
3. In the **Displays** panel, click **Display Editor**.
4. Click the **Display/View Config** tab.
5. In the **Display Configuration** section, select the This display is available in the solution panel to the left option.
6. In the **Display-Solution Configuration** section, select the user solution to associate with the current user display.
7. In the **Context View Configuration** section, select the associated check box.
This section allows the display/solution configuration that you created to be available in the **Show Compatible** section of the **Displays** panel.
8. You can select object types using one of the following approaches:



Note:

Context awareness occurs when you select the specified resource in the navigator. For example, if you associate a user display with the resource, `Type - Equipment Module`, then when an equipment module is selected in the navigator, the user display name in the solution panel is highlighted green. You can then select the highlighted user display to view it.

To...	Do this...
Select <u>individual</u> object types	<ul style="list-style-type: none"> a. From the Object Types list, select the object type you want to associate with the current user display, and then click the right arrow to add it to the Included Object Types list. Continue adding object types, as required. b. If applicable, add object types to the Excluded Object Types list.
Select <u>all</u> object types	<ul style="list-style-type: none"> a. Select the This display is compatible with all object types check box. Your display is shown under the Show Compatible section of the Displays panel for any selection in the navigator.

9. Click **Save**.

Custom (User-Created) Solutions

The **Display/View Config** tab is used to configure user-created solutions to be available in the **Displays** panel and to associate user displays with those user solutions.

The following table describes the available attributes.

Attributes	Value	Description
Display Configuration	User-defined	Select an option to indicate whether the selected user display is available in the solution panel.

Attributes	Value	Description
		<p>If you select This display is available in the solution panel, you must select a user solution from the list to associate the user display with.</p>
Context View Configuration check box	User-defined	<p>Select this check box to indicate that the current user display will be context aware. That is, when the user display is selected in the navigator, the compatible views are highlighted in the solution panel.</p> <p>This check box is available only when you select the Display Configuration option, The Display is available in the solution panel.</p> <p>If you clear this check box, the user display is still associated with the selected user solution; however, it is not context aware. That is, the display is not highlighted in the solution panel, nor does it appear in the Show Compatible list.</p> <div data-bbox="1024 1440 1421 1751" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: A user display can be associated with only one user solution, but can be associated with multiple object types.</p> </div>
Object Types	User-defined	Lists the object types you can associate with the current user

Attributes	Value	Description
		display. If the user display is already associated with an object type, that object type is not available to select from the list for this user display. Object types are available for selection only when the Context View Configuration check box is selected.
Included Object Types	User-defined	Lists the object types that the current user display is associated with.
Excluded Object Types	User-defined	Lists the object types that the current user display cannot be associated with.

Examples of Custom (User-Created) Solutions

This topic provides examples on creating user solutions.

Only included object types are context aware for the current user display. You can exclude object types to remove subsets of included object types. You can include individual resource types; however, each time a new resource type is added to the navigator that you want to include in context awareness, you must edit the user display.

Displays for Pieces of Equipment

You have a user display that can visualize properties on a piece of equipment, and you want that user display to work with all equipment-related resources, but not with non-equipment resources. In this case, you would include Equipment resources, but exclude the Area, Site, and Enterprise resource types.

Reporting Based on Resources

You want to create a financial report based on all physical equipment resources. Because control modules are representations of specific inputs and outputs and not physical equipment, you would include Equipment resources, but exclude the Control Module resource type.

SSL Encryption for SQL Server

SQL Server SSL Encryption

It is important to enable SSL encryption on your SQL Server in order to secure data. All connections for the Workflow database are encrypted, as well as any other databases on the same SQL server.

Performance Impacts

According to Microsoft[®], enabling encryption causes a 2-4% performance penalty.

SQL Server Certificate Requirements

This section describes the certificate requirements required for your SQL server.

SSL certificates must be:

- Valid.
 - The date and time of the current system must be between the certificate's **Valid From** and **Valid To** properties.
- Issued for server authentication.
- Placed in the certificate store of the local computer.
- The Common Name (CN) in the **Subject** property of the certificate must be the same as the fully qualified domain name (FQDN) of the server computer.



Note:

It is not recommended that you use self-signed certificates in a production environment.

Install a Certificate on a SQL Server

You can install a SSL certificate on your SQL server.

About this task

For more information about adding a certificate to a computer, see your Windows documentation.

Procedure

1. From the **Start** menu, click **Run**.
The **Run** dialog box appears.
2. In the **Open** field, enter **MMC**, and then click **OK**.
The MMC console appears.
3. From the **File** menu, click **Add/Remove Snap-in**.

4. Select **Certificates**, and then click **Add**.
The **Certificates snap-in** dialog box appears.
5. Select **Computer account**, click **Next > Finish** and then click **OK**.
6. Expand **Certificates (Local Computer) > Personal > Certificates**.
7. Right-click **Certificates**, select **All Tasks**, and then click **Import**.
The **Certificate Import Wizard** appears.
8. Use the **Import Wizard** to add a certificate to your computer.
9. Close all open windows.

Related reference

[SQL Server Certificate Requirements \(on page 321\)](#)

Enable SQL Server SSL Encryption

On your SQL server, you can enable SSL encryption.

Before you begin

About this task

**Note:**

According to Microsoft[®], enabling encryption causes a 2-4% performance penalty.

Procedure

1. From the **Start** menu, go to **All Programs > <Your SQL Server > Configuration Tools > SQL Server Configuration Manager**.
2. Expand **SQL Server Network Configuration**, right-click **Protocols for <Server Instance>**, and then select **Properties**.
The **Protocols for <Instance Name> Properties** dialog box appears.
3. Click the **Certificate** tab, and from the **Certificate** box drop-down menu, select the certificate that you want to work with, and then click **OK**.
4. Click the **Flags** tab, and in the **Force Encryption** field, select **Yes** from the drop-down menu, and then click **OK**.
5. To restart the SQL Server service, from the **Start** menu, point to **Administrative Tools**, and then click **Services**.

Related reference

[SQL Server Certificate Requirements \(on page 321\)](#)

Troubleshooting Certificates

Various issues may occur while enabling SSL encryption.

Certificate Is Not Available

For more information on resolving this issue, see the **Troubleshooting** section on the Microsoft® website: <https://support.microsoft.com/en-us/kb/316898>.

SQL Service Does Not Start after Enabling SSL Encryption

The following error messages may be written to the SQL Server Error log file:

- DateTime Server Unable to load user-specified certificate. The server will not accept a connection. You should verify that the certificate is correctly installed. See "Configuring Certificate for Use by SSL" in Books Online.
- DateTime Server Error: 17182, Severity: 16, State: 1.
- DateTime Server TDSSNIClient initialization failed with error 0x80092004, status code 0x80.
- DateTime Server Error: 17182, Severity: 16, State: 1.

Check to ensure that:

- The certificate itself is valid and has not expired.
- The certificate hash matches the certificate associated with the instance, but it was not loading.

For more information on resolving this issue, review the following websites:

- <https://support.microsoft.com/en-ca/kb/928779>
- <https://thesqldude.com/2011/08/03/sql-server-service-does-not-start-after-enabling-ssl-encryption/>

Certificate Handling

OPC UA Certificate Handling

An initial attempt to connect securely to an OPC UA server from the Workflow server will fail due to a missing certificate with the error: *The OPC UA server does not have any secure endpoints or the Workflow certificate is not trusted.*

A self-signed certificate can be created using the **Configure OPC UA** tool. Once a certificate is created, you can make a connection to the OPC UA server. The new certificate is sent to the server as part of its connection request. The OPC UA server returns its server certificate during the connection exchange.

The OPC UA server accepts the Workflow client certificate but does not initially trust it, placing it into the OPC UA server's untrusted or rejected certs folder. On the Workflow server, from the **Certificates Trust List** tab, run the **Configure OPC UA** tool. Select the certificate, and click **Trust** to trust this certificate.

To complete a secure connection, a manual process to trust the Workflow certificate must be executed on the OPC UA server.

Workflow (OPC UA client) Certificate Paths

Workflow saves its own certificate in: `C:\ProgramData\Proficy\OpcUaCertificate\own\certs`

The Workflow certificate file will be named similar to this: **Workflow [821C3D37F1B91FB6FF87EAA8FA9116B5F4526F51].der**

The OPC UA server certificate on Workflow is saved in: `C:\ProgramData\Proficy\OpcUaCertificate\trusted\certs`

In order for Workflow to connect securely to an OPC UA server, the Workflow client certificate must be moved from the server's untrusted or rejected certs folder to the server's trusted folder. Instructions on how to do this are described in **Move Untrusted Workflow Certificates on the OPC UA Server**.

Related information

[Move Untrusted Workflow Certificates on the OPC UA Server \(on page 325\)](#)

[Modify Server Certificate Folder Location \(on page 328\)](#)

Generate OPC UA Self-Signed Certificates Manually

You can use the OPC UA Configuration tool to manually generate a self-signed Workflow security certificate.

About this task

Once generated, a self-signed Workflow security certificate is untrusted and must be moved to the trusted store on the OPC UA server. See **Move Untrusted Workflow Certificates on the OPC UA Server** for more information. The Workflow security certificate is stored in the `C:\ProgramData\Proficy\OpcUaCertificate\own\certs` folder.

Procedure

1. From the OPC UA server, click **Start > All Programs > General Electric > Workflow > Configuration > Configure OPC UA**.

The **Configure OPC UA** tool appears.

2. Click **Generate Self-Signed**.

The following actions occur:

- Configuration saved.
- Created <Workflow certificate name>.
- Skipping: Certs do not register with LDS.

Related information

[Move Untrusted Workflow Certificates on the OPC UA Server \(on page 325\)](#)

Move Untrusted Workflow Certificates on the OPC UA Server

You can manually move certificates from the OPC UA server's untrusted or rejected certificates folder to its trusted certificate folder.

About this task

Moving the Workflow certificates into the trusted certificate folder allows a secure connection between the OPC UA server and the Workflow server. Some OPC UA servers provide a configuration UI to facilitate moving certificates, but for other servers, you will need to move the certificates manually.

OPC UA Server Certificate Paths

The following paths are examples of a typical OPC UA server. OPC UA server certificate paths may vary.

The OPC UA server %INSTALLDIR% default directory is: C:\Program Files (x86)\Proficy PROJECTNAME\

The OPC UA server rejected certs folder is: %INSTALLDIR%\projects\%PROJECTNAME%\pki
\rejected

The OPC UA server trusted certs folder is: %INSTALLDIR%\projects\%PROJECTNAME%\pki
\trusted\certs



Tip:

To determine the filename of the certificate to move, look for the Workflow **.der** certificate in the C:\ProgramData\Proficy\OpcUaCertificate\own\certs folder on the Workflow server. The certificate file name will be prefixed with **Workflow** and look similar to this: **Workflow [821C3D37F1B91FB6FF87EAA8FA9116B5F4526F51].der**

Procedure

1. From the OPC UA server, in the reject folder, find the cert **.der** file with the same name without the Workflow prefix. (e.g., **821C3D37F1B91FB6FF87EAA8FA9116B5F4526F51.der**)
2. Move the selected certificate from the %INSTALLDIR%\projects\%PROJECTNAME%\pki\rejected folder to the %INSTALLDIR%\projects\%PROJECTNAME%\pki\certs\trusted folder.



Note:

The paths shown are examples. Your OPC UA untrusted and trusted folder paths may vary.

What to do next

Attempt a connection from the Workflow server to the OPC UA server.

Related information

[Modify Server Certificate Folder Location \(on page 328\)](#)

Register Workflow Certificates with Global Discovery Server (GDS)

You can use the OPC UA Configuration tool to connect to the Global Discover Service (GDS) server, sign a self-signed Workflow security certificate, and automatically register the application with GDS.

About this task

Registering the Workflow security certificate with GDS makes the certificate trusted by other applications which are registered in GDS and allows you to add a CA certificate in Workflow.

Other Actions

You can perform the following other actions to the selected certificate from the **Certificates Management** tab:

- Click **Update Trust List** to replace the trust certificates with the ones provided by the GDS.

You can perform the following other actions to the selected certificate from the **Certificates Trust List** tab:

- Click **View** to view information about the certificate in the **Certificate Details** dialog box.
- Click **Add** to select a certificate to trust from a folder.
- Click **Make Issuer** to make a CA certificate untrusted, but still trust the certificate signed by this CA certificate.

- Click **Reject** to make the selected certificate untrusted. The certificate is moved to the C:\ProgramData\Proficy\OpcUaCertificate\pki\certs\rejected folder.
- When a certificate is rejected, you can click **Trust**, then click **Yes** in the **Trust Certificate** dialog box to make the selected certificate trusted again.
- Click **Delete** to remove the selected certificate from the C:\ProgramData\Proficy\OpcUaCertificate\pki\certs\trusted folder.

Procedure

1. From the OPC UA server, click **Start > All Programs > General Electric > Workflow > Configuration > Configure OPC UA**.

The **Configure OPC UA** tool appears.

2. Click **Configure GDS**.

The **Select Certificate Management Service (GDS)** dialog box appears.

3. Enter the following information:

- a. Enter an endpoint URL for the GDS server into the **Endpoint URL** field.



Note:

You can click **Search for Endpoints** and select from a list of endpoint URLs.

- b. Enter a user name for the GDS server in the **User Name** field.
- c. Enter a password for the selected user account in the **Password** field.
- d. Click **OK**.

A connection is made to the GDS server.

4. Click **Generate Self-Signed**.

The following actions occur:

- Configuration saved.
- Created <Workflow certificate name>.
- Skipping: Certs do not register with LDS.

The Workflow security certificate is stored in the C:\ProgramData\Proficy\OpcUaCertificate\own\certs folder but is untrusted.

5. Click **Request from GDS**.

The following actions occur:

- Configuration saved.
- Selected GDS at <certificate directory used>. Configure GDS appears if GDS is not configured yet or the GDS connection has failed.
- Registered <application certificate>.
- Sent certificate request.
- Checking if certificate request has completed.
- Replaced trusts list with the one provide by the certificate directory.
- Skipping: Certs do not register with LDS.

The Workflow security certificate is moved to the `C:\ProgramData\Proficy\OpcUaCertificate\pki\certs\trusted` folder and is now trusted.

What to do next

You can now add OPC UA through the Workflow client. See **Add an OPC UA Server**.

Generate Certificates for Workflow 2.6

Workflow 2.5 and earlier does not support the creation of self-signed certificates with the Subject Alternative Name (SAN) property.

Certificates missing the SAN property are considered to be insecure. If the SAN is missing, Chrome v58.x and later responds with an error when you attempt to connect. For more information, see <https://developers.google.com/web/updates/2017/03/chrome-58-deprecations> and access the **Remove support for commonName matching in certificates** section.

If Workflow 2.6 is installed on Windows 10 or Windows Server 2016, it generates self-signed certificates using Powershell 5 New-SelfSignedCertificate cmdlet so a SAN can be specified. Workflow 2.6 installations on older operating systems will continue to generate certificates using the previous method. These certificates do not have the SAN property.

Modify Server Certificate Folder Location

You can modify the location of the Workflow server certificates folder.

About this task

Use the `%INSTALLDIR%\Proficy\Proficy Workflow\ProficyServer.exe.config` file to edit the location of **Certificates** and **Trust Lists**.

Procedure

1. On the Workflow server, open the **ProficyServer.exe.config** file.
2. Find the **UaApplicationConfiguration** section.

3. Change the specified file path to relocate the server certificates.
4. Save and exit the file.

UaApplicationConfiguration in ProficyServer.exe.config

```
<UaApplicationConfiguration>
  <SecuredApplication xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://opcfoundation.org/UA/2011/03/SecuredApplication.xsd">
    <!-- Specify location of Certificates and Trust Lists -->
    <ApplicationCertificate>
      <StoreType>Directory</StoreType>
      <StorePath>%CommonApplicationData%\Proficy\OpcUaCertificate\own</StorePath>
      <SubjectName>CN=Workflow/O=GeneralElectric/DC=localhost</SubjectName>
      <ValidationOptions>0</ValidationOptions>
    </ApplicationCertificate>
    <TrustedCertificateStore>
      <StoreType>Directory</StoreType>
      <StorePath>%CommonApplicationData%\Proficy\OpcUaCertificate\trusted</StorePath>
      <ValidationOptions>0</ValidationOptions>
    </TrustedCertificateStore>
    <IssuerCertificateStore>
      <StoreType>Directory</StoreType>
      <StorePath>%CommonApplicationData%\Proficy\OpcUaCertificate\issuers</StorePath>
      <ValidationOptions>0</ValidationOptions>
    </IssuerCertificateStore>
    <RejectedCertificatesStore>
      <StoreType>Directory</StoreType>
      <StorePath>%CommonApplicationData%\Proficy\OpcUaCertificate\rejected</StorePath>
      <ValidationOptions>0</ValidationOptions>
    </RejectedCertificatesStore>
  </SecuredApplication>
</UaApplicationConfiguration>
```

Resolve Port conflicts between prior Proficy Workflow installations and Web HMI

For existing Workflow installations prior to 2.6, Workflow and Web HMI ports are both set to 8444, making applications installed on the same machine incompatible.

About this task

To resolve this issue, you will need to change the default port for your Workflow REST services from 8444 to 8847 or provide another port as part of your Workflow server information in Web HMI.

**Note:**

For Workflow 2.6, the default port for REST services hosted by Proficy Workflow is changed from 8444 to 8447.

Provide the **Server Name** in the format <WFServer>:<Port>. If no port information is provided, the default is used.

The REST service port can be changed on the Workflow server using the following steps:

Procedure

1. Open the file **ProficyServer.exe.config** and locate the entry for `RestHostConfiguration`.
2. In the `baseAddress` attribute, adjust the port the desired value, e.g., for port 8447.

```
baseAddress="https://localhost:8447/ProficyWebApi."
```

The new port must be configured with a valid SSL certificate.

**Note:**

All Workflow 2.6 installs will have port 8447 configured using the Workflow SSL certificate.

3. See [How to Configure a Port with an SSL Certificate](#) for information about using a different port.
4. Using the **Local Machine Certificates** snap-in of **MMC.exe**, locate the SSL certificate you want to use. This can be a certificate bound to other Workflow ports, such as 8443.
5. View **Details** to determine its thumbprint.
6. Use the thumbprint as the `certhash` value in the `netsh.exe` command below:

```
netsh.exe http add sslcert ipport=0.0.0.0:<portnumber> certhash=<thumbprint>  
appid={00112233-4455-6677-8899-AABBCCDDEEFF}
```

Chapter 6. Security

Security Overview

Workflow Security

Security configuration and maintenance is performed by Workflow system administrators.

In some companies, access to the production configuration and management environment is available to everyone. In such an environment, changes to the data files and access to files and applications are not critical to the process. However, in other companies, these applications and data are critical to the process and, thus, available only to authorized personnel.

Workflow provides an integrated security program to assist you in protecting your process.

Understanding Security

Your main design goal when developing a Workflow security strategy is to effectively use the Workflow security configuration.

Workflow simplifies security configuration by abstracting the three main areas that are constantly changing in a plant or facility. These main areas are:

- people/personnel
- equipment/machinery
- operational hierarchy of the people

New personnel are hired and some retire in an organization. Machinery is added, replaced, and updated. People are promoted; for example, from being an operator to a supervisor, and so on.

Workflow security accommodates all of these cases using three different constructs.

- People can be configured in groups; for example, Production Group, Maintenance Group.
- Machinery is modeled with the Equipment model using the S95 Model Editor, which defines a plant in a hierarchical fashion. An Enterprise can have multiple sites and sites can have multiple areas, and so on.
- An operational hierarchy can be defined as a set of keys that each person can hold. For example, an Operator can have keys to Start and Stop a workflow. A Supervisor can have additional keys to Create and Delete a workflow.

The first step in configuring security is to add people (users) to the system, then configure groups, and then define the key sets. Only an Administrator user can define these three fundamental pieces of security configuration. An Administrator user is created when Workflow is installed.

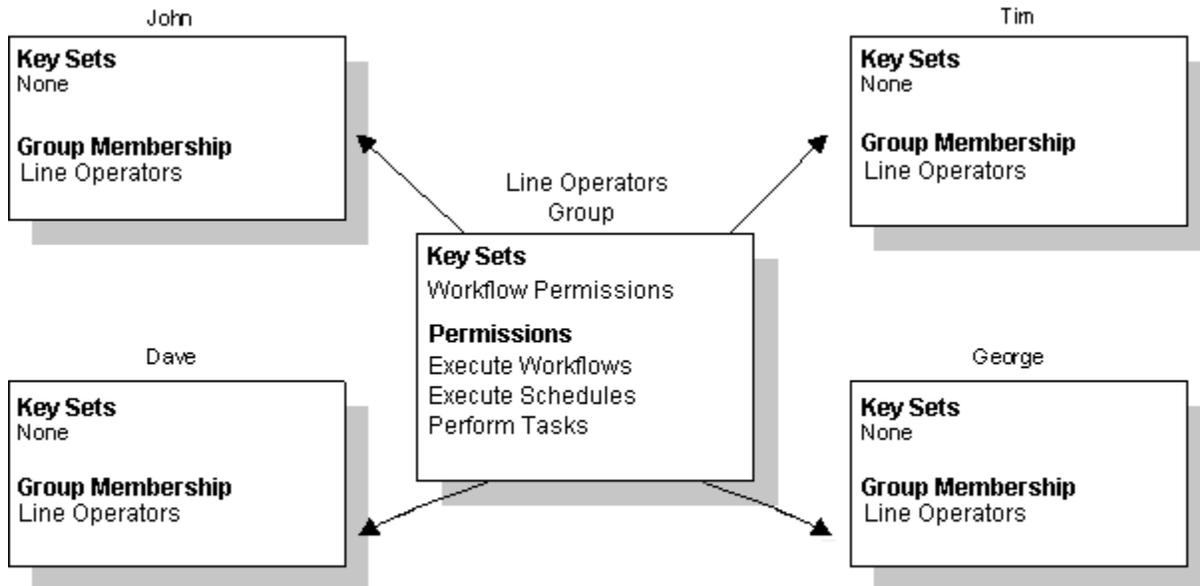
After the groups and key sets are defined, security can be applied to any resource in the system. For example, a group can be granted a key set on a resource such as an Enterprise in the Equipment model. Any user who belongs to the configured group will then, upon logging in to the Workflow client, have the rights granted by the key set on the Enterprise.

Using groups minimizes the amount of work needed to create similar users while providing you with flexibility and power. For example, when a new user comes in the plant, the user can simply be added to an existing group and will automatically inherit all the rights defined for that group.

In the following example, John, Dave, Tim, and George are all line operators. Their needs are summarized in the table.

User Name	Key Set	Permissions
John	Workflow Permissions	Execute Workflows, Execute Schedules, Perform Tasks
Dave	Workflow Permissions	Execute Workflows, Execute Schedules, Perform Tasks
Tim	Workflow Permissions	Execute Workflows, Execute Schedules, Perform Tasks
George	Workflow Permissions	Execute Workflows, Execute Schedules, Perform Tasks

Since each line operator requires access to the same permissions, it is possible to create a group called Line Operators that provides these privileges. After you create the group, you can assign it to each line operator, as the following figure shows.



System Protection

There are different levels of security that you can implement to protect your system. On one level, you can control the security of your machines and buildings, as well as your processes. On another level, you can implement security for your operating system and your network using firewalls, passwords, and filters.

You can also protect your data from unauthorized changes by restricting access to your system. The information in this section focuses on the security in Workflow, not your operating system or network.

Logging into Workflow requires a login name and password. Depending on your configuration, this data can be the same or separate from your Windows® login name and password. For more information on Windows login, refer to the **Windows Users** section.

Security is user-based, meaning operators cannot access anything unless you assign access to the specific areas.

Workflow security is flexible and easy to use, allowing you to assign operator rights, login names, and passwords. Refer to **Users and Groups** for more information about configuring your users, groups, and key sets.

Related information

[Windows Domain-Based Security \(on page 357\)](#)

[Workflow Users and Groups \(on page 338\)](#)

Security Concepts

Before you restrict access to Workflow features, you need to understand how security works.

The security concepts described here are described in more detail in **Understanding Security**. For information on implementing the concepts, see the **Workflow Users and Groups** section.

- **People** - allow you to define security credentials and properties that are specific to an individual user. Security credentials and properties are defined in the S95 Model Editor.
- **Key Sets** - allow you to assign permissions to users who hold a specific key set. In addition, key sets inherit permissions from their child sets. Workflow provides the following predefined high level key sets:
 - Guest
 - S95 Configurator
 - Workflow Author

The S95 Configurator key set includes the following child key sets:

- Equipment Configurator
- Material Configurator
- Personnel Configurator

The Workflow Author key set includes the following child key sets:

- Workflow Builder
- Workflow Manager
- Workflow Supervisor
- Workflow Operator

You can also create your own key sets at any level.

- **Groups** – allow you to create groups of users who share the same key sets. Workflow provides the predefined groups Administrators, Guests, and Windows Users.
- **Client Access** – allows you to configure automatic login and logout, lockdown mode, default startup options for an individual workstation, and user overrides.

Related information

[Understanding Security \(on page 331\)](#)

Security Status

Workflow security is always enabled. This prevents any unauthorized users from accessing or modifying any part of the system.

With security, all users must log in with their user accounts to gain access to the areas they have been given permissions to.

Design and Plan Security

Security is enabled when you install Workflow and cannot be disabled; therefore, you should plan and design all required users and groups before you deploy Workflow.

When you install Workflow, you are prompted to create an Administrator user. This allows you to log in as the administrator and begin creating users and groups. Workflow provides default groups and key sets that you can examine to learn how to create your own users and groups.

After you have logged into Workflow as the newly created Administrator user, you can create one or more additional users who have a key set assigned to them that allows them to configure personnel resources for your facility. You can also create users who have a key set assigned that allows them to configure security aspects, such as configuring and assigning groups and/or key sets. These can be the same or different users, but if they will have security privileges, they must be accounted for in your security plan.

Your security design plan should include all of the information that you need to configure your personnel resources. For example, if you have a set of users that all perform the same tasks, you can create a group and add each of those users to the group. You can then assign key sets to the group and all members of the group inherit the permissions configured for those key sets.

By designing a well-defined security plan, you can build a security system that provides all of the permissions and access that your employees need to perform their assigned facility operations.

For information on logging into Workflow, see **Login and Logout**.

Related reference

[Automatic Login/Logout Configuration \(on page 393\)](#)

Related information

[Key Sets \(on page 371\)](#)

Encrypted Communication Between SQL Server and Workflow Server

For a secure system, GE Digital recommends that you encrypt the communication between the application server and the SQL Server.

If communications between these servers are not encrypted, certain tools can read the plain text SQL queries and data transmitted between those servers.

For information about encrypting connections to SQL Server, see <http://msdn.microsoft.com/en-us/library/ms189067.aspx>.

For information about enabling SSL encryption for an instance of SQL Server, see <http://support.microsoft.com/kb/316898>.

For information about configuring certificates for use by SSL, see <http://msdn.microsoft.com/en-us/library/ms186362.aspx>.

Security Authorization

Authorization is a security feature that gives particular users permission to access data and leverage capability with the Vision application.

Foundation of Authorization Implementation

The foundation of this system is the concept of roles and the hierarchy within those roles. You create roles and then create a cascading system of permissions (or privileges) based on those roles. When you first create a role, you must determine where within your system that role will act. For example, you determine which pieces of equipment that a person in that role can use. Then, you determine what the role is going to be able to do, as well as who is going to be able to perform that action. By doing this, you decide the precise actions that a particular person or group of people will be able to do within your application.

Authorization in Vision

A person or a group of people (users) can be assigned a role that allows them to use a resource (for example, a piece of equipment) to perform an action within the limits of their access (privilege), which was defined for the role assignment.

Central Concept	Terms	Description
>Role Assignment	>User, User Group	The person, such as an operator or supervisor, who is assigned to perform particular actions within predefined privileges. A group is a collection of users.
	>Resource, Resource Set	The scope of the assignment, such as to what or to whom a privilege is assigned. For exam-

Central Concept	Terms	Description
		ple, a piece of equipment or access to a system.
	>Privilege, Privilege Set	The ability to perform operations.

Key Features and Terms

Vision uses a role-based access control security system to restrict application access to authorized users. This authorization system is the next generation of authorization. Systems like Workflow and Historian will evolve to this model over subsequent releases.

Currently, there are differences between Vision and Workflow. The following table highlights the distinction in terminology between systems.



Note:

For more information about authorization and how to use it, see the **Vision** documentation.

Vision Term	Workflow Term	Implementation
User, User Group	Users, Groups	Users and groups are shared between both systems. You can manage them in each application.
Privilege (privilege set)	Permission	Privileges in Vision are not visible or managed in Workflow.



Note:

User and group resources are available in both applications; however, if you delete this resource in Vision, you must not delete it in Workflow if you still require this resource to perform actions within this application.

Vision Term	Workflow Term	Implementation
Resource (resource set)	Any resource; for example, equipment	Resource sets given to a Vision user (?) are not enforced in Workflow.
Assignment	Key sets	Assignments granted in Vision are not enforced in Workflow.

Users and Groups

Workflow Users and Groups

Personnel are the users required for your facility and processes, including security set up.

Personnel consists of classes, groups, key sets, and people. The following table provides a brief description of each of these. For more information on each, refer to the specific overview topic.

Folder	Description
Classes	Groups of common properties between users.
Groups	Collections of similar users that are configured to have the same access rights.
Key Sets	Groups of permissions that define the operations a user or group of users can perform.
People	The users that are configured to have access to your facility and processes.

Users are assigned to different aspects of your facility, such as production and workflow resources, and are also used extensively in security. Key sets and groups are used exclusively for security.

If you have a specific set of users that require the same properties, such as shift schedules, then creating a personnel class that includes this information makes configuring those users easier. You can then apply the class to all of the applicable users. The properties configured for the class are inherited by those users.

Before configuring your personnel model, it is best to create a design plan that describes all aspects of your facility and the areas that require users, including a security plan.

An important aspect of the personnel model is the ability to assign security and personnel configuration permissions to specific users. By limiting these permissions to a subset of users, you can feel more assured that your personnel and production security configurations are accessible only by those with the rights to use them.

You can create as many or as few personnel resources as required for your facility operation. Workflow provides default groups and key sets that you can use for your personnel model, or you can create your own.

**Important:**

The Administrators group must have at least one user as a member at all times. If you have only one administrator and that user changes, you can edit the user's name and description, as well as the security credentials.

**Note:**

When you install Workflow, you are prompted to create an Administrator user. This allows you to log in as the administrator and begin creating Workflow users and groups. Workflow provides default groups and key sets that you can examine to learn how to create your own users and groups.

Workflow Users and Personnel Classes

People are the users that are configured to have access to your facility's processes.

Workflow identifies each user with a login name, a password, and an optional e-mail address. Users can belong to one or more groups. When a user belongs to a group, he or she inherits all the key sets assigned to the group. The user can have separate key sets in addition to the group key sets.

When you are designing your security plan, always include the full name, login name, and password for each user. If you plan to use Windows security, you should also include the domain name if you want to store the user information on a domain controller.

User information is configured using both the **Security Editor** and the **S95 Model Editor**.

Personnel Classes

A personnel class is a group of properties that are common between users. For example, you may have an *operator* class that contains properties related to shift schedules or machine qualifications. Personnel classes can be applied only to users and multiple classes can be applied to a given user.

Add Users and Personnel Classes

Users are the people who have access to various areas of your facility. Personnel classes are used to configure common user properties.

About this task

The following steps outline the process for adding users whose Workflow security credentials you plan to manage strictly within the program. You can also use Windows domain user accounts to manage access to and within Workflow. For more information, see **Windows Domain-based Security**.

Procedure

1. In the navigator, click **Personnel**.
2. In the **Folders** list, select the resource type you want to work with.



Tip:

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. Click **Add**.
4. In the **Name** and **Description** fields, enter a name and description for the resource.



Note:

There may be limits to the type and number of characters you can use.

5. Click **OK**.

What to do next

After creating a Workflow user, you must configure the user's security credentials; otherwise, the user will not be able to log into Workflow.

Related information

[Convert between Windows domain and Workflow users \(on page 368\)](#)

[Windows Domain-Based Security \(on page 357\)](#)

[Configure Security Credentials for a User \(on page 342\)](#)

Configure Users and Personnel Classes

Properties are used to define the unique attributes of your users and personnel classes.

Procedure

1. In the navigator, click **Personnel**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, click **S95 Model Editor**.
4. Click the **Properties** tab, and then click **Add Property**.
5. In the **Name** and **Description** fields, enter a name and description for the property/parameter.
6. From the **Data Type** list, select a data type.
7. In the **Value** field, enter a value for the selected data type.
8. **Optional:** In the **Unit of Measure** field, enter a unit of measure for the selected data type.
9. Click **Save**.

Use Custom Property Types in the Personnel Model

Procedure

1. From the navigator, click **Personnel**.
2. From this list, select the resource you want to work with.
3. In the Displays panel, click **Production Management > S95 Model Editor**.
4. Click the **Properties** tab, and then click **Add Property**.
5. In the **Name** field, enter a name for this property.
6. From the **Type** field, select a custom property type.
7. In the **Value** field, select a value for the custom property type.
8. **Optional:** If applicable, in the **Unit of Measure** field, enter a unit of measure for the custom property type.
9. Click **Save**.

Associate a Personnel Class with a User

If you have defined a personnel class with a set of properties that are common to a specific group of users, you can associate that class with the users to make property configuration for those users easier.

Procedure

1. In the navigator, click **Personnel > People**.
2. In the **People** list, select the user you want to associate a personnel class with.
3. In the **Displays** panel, click **S95 Model Editor**.

4. Click the **Classes** tab, and then click **Add Class**.
The Universal Browser appears.
5. Locate and select the personnel class you want to add, and then click **OK**.



Tip:

To remove the personnel class you added, select it, and then click Delete Selection.

6. Click **Save**.

Configure Security Credentials for a User

About this task



Note:

You can edit a user's security credentials at a later time from either the **S95 Model Editor** or from the **Security Editor**.

Procedure

1. In the navigator, click **Personnel > People**.
2. In the **People** list, select the user you want to configure security credentials for.
3. In the **Displays** panel, click **S95 Model Editor**.
4. Click the **Details** tab.
5. **Optional:** In the **Full Name** field, enter the user's name.
This name is for informational purposes in electronic signatures only. It is not a unique identifier and is not used for security credentials.
6. Click **Create Account**.
The **Credentials** area expands, displaying further input fields
7. In the **Login Name** field, enter a unique user name for the user you are configuring.
8. In the **Password** field, enter the password for the user you are configuring.
9. In the **Confirm Password** field, re-enter the password.
10. **Optional:** In the **Email** field, enter the e-mail address of the user you are configuring.
11. Click **Save**.
This user is now active.

Reactivate Locked Out User Accounts

If a Workflow user is locked out due to entering incorrect login credentials, you can reactivate the account so that the user does not have to wait for the lockout duration to pass.

Procedure

1. In the navigator, click **Personnel > People**.
2. In the **People** list, select the user whose account you want to reactivate.
3. In the **Displays** panel, click **S95 Model Editor**.
4. Click the **Details** tab.
5. Click **Reactivate Account**.
6. Click **Save**.

Add a User to One or More Groups

Use the **Security Editor** to add users to groups and to remove users from groups.

Procedure

1. In the navigator, click **Personnel > People**.
2. In the **People** list, select the user you want to add to a group.
3. In the **Displays** panel, click **Security Editor**.
4. In the **Security Groups** area, click **Add/Remove Groups**.
The **Add/Remove Groups** dialog box appears.
5. From the list of available groups, select the group(s) you want to add the user to, and then click the right arrow.

**Tip:**

- To add a user to multiple groups, press the **Ctrl** key, and then select all of the groups you want to add the user to, or click the double right arrow to add the user to all available groups.
- To remove a user from a group, from the list of groups the user is a member of, select the group you want to remove the user from, and then click the left arrow. Press the **Ctrl** key to select multiple groups, or click the double left arrow to remove a user from all the groups.

6. Click **OK**, and then click **Save**.

Related information

[Add One or More Users to a Group \(on page 353\)](#)

[Configure Security Credentials for a User \(on page 342\)](#)

Edit or View a Group for a User

Procedure

1. In the navigator, click **Personnel > People**.
2. In the **People** list, select the user whose group associations you want to view or edit.
3. In the **Displays** panel, click **Security Editor**.
4. In the **Security Groups** area, select the group you want to view or edit, and then click **Edit/View Group**.

The **Security Editor** displays the information for the selected group.

5. If applicable, make any required changes to the group.
6. Click **Save**.

Related information

[Edit or View Users That Belong to a Group \(on page 354\)](#)

[Add workstations to key sets for a group \(on page 374\)](#)

[Add key sets to a workstation for a group \(on page 375\)](#)

Remove a User Account

You can remove user accounts from Workflow; however, before you delete a user account ensure that it is not associated with any process in your system. If you delete a user account that is in use, you may encounter problems or unpredictable behavior from the application.

Procedure

1. In the navigator, click **Personnel > People**.
2. In the **People** list, select the user you want to remove from a group.
3. In the **Displays** panel, click **S95 Model Editor**.
4. Click the **Details** tab.
5. Click **Remove Account**.
6. Click **Save**.

Workflow User Account Details

The **Details** tab on the **S95 Model Editor** for Workflow users offers general information about the selected user and allows you to create and configure the login credentials for your Workflow users.

You can change the name and description of this resource.

Attribute	Value	Description
Name	Required; User-defined up to 50 characters	Specifies a display name for a resource.
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

Attribute	Value	Description
Full Name	User-defined up to 255 characters	Specifies the name of the user. <div data-bbox="1024 688 1419 1052" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: This name is for informational purposes in electronic signatures only. It is not a unique identifier and is not used for security credentials. </div>
Create Account	N/A	Click to create a new user account.
Remove Account	N/A	Click to delete an existing user account.

User Account Status

The account status area indicates whether the selected Workflow user's account is active or disabled. A disabled account is an account that has been locked out due to exceeding the lockout threshold defined for your system. Only an Administrator user, or a user with the required permission can reactivate a locked out user account.

Attribute	Value	Description
Active	N/A	Indicates that the current user account status is active.
Reactivate Account	N/A	Click to reactivate a locked out Workflow user account.

Security Credentials

The following table describes the user-specific attributes on the **S95 Model Editor** required to configure the security credentials for a user. These attributes are available only when the **Create Account** button is clicked.



Note:

You can also access the security credentials for the selected user by clicking the **Edit Security Credentials** link on the **Security Editor**.

Attribute	Value	Description
Login Name	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 20 characters	Specifies the unique login name for the selected user.
Password	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 255 characters	Specifies the unique password for the selected user.
Confirm Password	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 255 characters	Specifies the unique password for the selected user a second time.
E-mail	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 255 characters	Specifies the unique e-mail address for the selected user.

User and Personnel Class Properties

The **Properties** tab on the **S95 Model Editor** for Workflow users and personnel classes allows you to add, modify, and delete properties that are specific to a personnel class or person (user), or to view the properties for any personnel classes that are associated with the selected user.

The following table describes the information available on the Properties tab, where you can configure the values for the property attributes.

Attribute	Value	Description
List	User-defined	Displays a list of properties for the selected resource that can be arranged by name, type, and so

Attribute	Value	Description
		on, depending on the property. These properties can be inherited from another resource or created for the current resource.
Add Property	N/A	Click to add a new property.
Delete Property	N/A	Click to remove the selected property.
Name	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 50 characters	Specifies a display name for the selected property.
Description	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 255 characters	Provides more information about a property that a user may find helpful.
Data Type	System-defined	<p data-bbox="1019 934 1414 1018">Specifies the data type of the property.</p> <div data-bbox="1024 1052 1419 1881" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p data-bbox="1040 1073 1170 1115"> Note:</p> <ul data-bbox="1166 1163 1403 1822" style="list-style-type: none"> <li data-bbox="1166 1163 1403 1325">• This property cannot be changed if it is inherited from another resource. <li data-bbox="1166 1346 1403 1640">• When you select Browse for class, the Universal Browser appears where you can select an equipment class. <li data-bbox="1166 1661 1403 1822">• This field is not available if the Linked check box is selected. </div>

Attribute	Value	Description
Value	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 255 characters	Specifies the value for the selected data type. Valid values depend on the selected data type.
UoM (Unit of Measure)	User-defined alphanumeric string (A-Z, a-z, 0 - 9) up to 20 characters	Specifies the unit of measure for the data type, such as gallons, inches, tons.

User and Personnel Class Associations

The **Classes** tab on the **S95 Model Editor** for Workflow users allows you to associate one or more personnel classes with the selected user.

A personnel class is a group of properties that are common between users. For example, you may have an operator class that contains properties related to shift schedules or machine qualifications. Personnel classes can be applied only to users and multiple classes can be applied to a given user.

Associate a User with a Running Workflow

Using personnel classes, you can configure the user interface of the Web Task List by associating users to a class.

User Security Configuration

When configuring user accounts, the **Security Editor** is used to add users to groups, as well as to view and edit the key sets assigned to any groups the user is a member of.

Security Groups

The following table describes the information available in this section:

Attribute	Value	Description
Group List	User-defined	Displays the groups the current user is a member of.
Edit/View Group	N/A	Click to display the selected group's information in the Security Editor . You can view or edit the group's information.

Attribute	Value	Description
Add/Remove Groups	N/A	Click to open the Add/Remove Groups dialog box.
Add/Remove Groups dialog box: Available Groups	User and system-defined	Displays the groups that are available to add the selected user to.
Add/Remove Groups dialog box: Group List	User-defined	Displays the list of groups that the selected user is already a member of.

Key Sets

The following table describes the read-only information in this section:

Attribute	Value	Description
Display Options	User-defined	Select the method you want to use to display the information for the key sets assigned to the user. You can display by key set, by resource, or by workstation.
Resource	User-defined	Displays all of the resources that the selected user holds key sets for.
Workstation	User-defined	Displays all of the individual workstations, including All Workstations if applicable, that the selected user holds key sets for.
Group	User-defined	Displays the groups that the selected user is a member of that holds key sets for the resource. This column displays when you select either the Show by Resource or Show by Workstation option.

Attribute	Value	Description
Key Set	User-defined	Displays the key sets that the selected user holds.
Edit/View Resource	N/A	Click to display the selected resource's information in the Security Editor . You can view or edit the resource's information.

Personnel Class Details

The **Details** tab on the **S95 Model Editor** for personnel classes offers general information about the selected personnel class.

You can change the name and description of this resource.

Attribute	Value	Description
Name	Required; User-defined up to 50 characters	Specifies a display name for a resource.
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

Personnel Class Properties

The **Properties** tab on the **S95 Model Editor** for personnel classes allows you to add, modify, and delete properties that are specific to the selected personnel class.

The following table describes the information available on the Properties tab, where you can configure the values for the property attributes.

Attribute	Value	Description
List	User-defined	Displays a list of properties for the selected resource that can be arranged by name, type, and so on, depending on the property. These properties can be inherited

Attribute	Value	Description
		from another resource or created for the current resource.
Add Property	N/A	Click to add a new property.
Delete Property	N/A	Click to remove the selected property.
Name	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 50 characters	Specifies a display name for the selected property.
Description	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 255 characters	Provides more information about a property that a user may find helpful.
Type	System-defined	<p data-bbox="1024 852 1416 930">Specifies the data type of the property.</p> <div data-bbox="1024 961 1416 1791" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p data-bbox="1040 982 1170 1024"> Note:</p> <ul data-bbox="1170 1073 1399 1734" style="list-style-type: none"> <li data-bbox="1170 1073 1399 1234">• This property cannot be changed if it is inherited from another resource. <li data-bbox="1170 1255 1399 1549">• When you select Browse for class, the Universal Browser appears where you can select an equipment class. <li data-bbox="1170 1570 1399 1734">• This field is not available if the Linked check box is selected. </div>

Attribute	Value	Description
Value	User-defined alphanumeric string (A-Z, a-z, 0 - 9) up to 8000 characters	Specifies the value for the selected data type. The options available depend on the selected data type.
UoM (Unit of Measure)	User-defined alphanumeric string (A-Z, a-z, 0 - 9) up to 20 characters	Specifies the unit of measure for the data type, such as gallons, inches, tons.

Workflow Groups

Workflow provides the ability to categorize your users into groups. Groups allow you to assign common permissions required by two or more users who perform similar tasks, and provide you with a simpler approach to implementing security.

Groups are collections of similar users that are configured to have the same access rights to your facility's processes. When you assign key sets to a group, each member of the group inherits those key sets.

Key sets are assigned to groups only; you cannot assign key sets to individual users. This assigns common key sets needed by two or more users performing similar tasks. Configuring your groups in this way provides a modular approach that is easy to maintain.

You can simplify security configuration significantly if you take the time to assess your users' needs. If the security requirements at your site do not warrant such an effort, use the default groups provided, which define functional roles in a manufacturing facility. The default groups are:

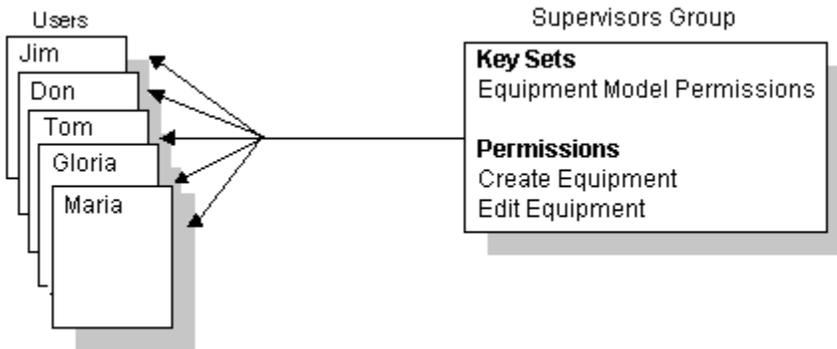
- Administrators
- Guests
- SSO Users
- Windows Users



Note:

- The SSO Users group should be used if you are using GE Single Sign On security.
- The Windows Users group should be used if you are using Windows authentication.

For example, in the following figure, the Supervisors group defines access to the equipment model to create and edit equipment resources. These permissions define the common security rights shared by all users in the group.



Related information

[Configuring Domain-Based Access Control to Workflow Functionality \(on page 361\)](#)

Add One or More Users to a Group

Use the **Security Editor** to add one or more users to a group and to remove users from groups.

Before you begin

You must have configured a user name and password for the selected user before you can add the user to a group.

Procedure

1. In the navigator, click **Personnel > Groups**.
2. From the list of groups, select the group you want to add members to.
3. In the **Displays** panel, click **Security Editor**.
4. Click the **Workflow Users** tab.
5. Click **Add/Remove Groups**.
The **Add/Remove Groups** dialog box appears.
6. From the list of available members, select the user(s) you want to add to the group, and then click the right arrow.

**Tip:**

- To add multiple users to the group, press the **Ctrl** key and select all of the users you want to add to the group, or click the double right arrow to add all available members to the group.
- To remove a user from a group, from the list of members, select the user you want to remove, and then click the left arrow. Press the **Ctrl** key and select multiple members, or click the double left arrow to remove all members from the group.

7. Click **OK**, and then click **Save**.

Related information

[Add a User to One or More Groups \(on page 343\)](#)

Edit or View Users That Belong to a Group

Use the **Security Editor** to view or edit a group member's security information.

Before you begin

You must have added one or more users to the group before you can edit or view a user.

Procedure

1. In the navigator, click **Personnel > Groups**.
2. In the **Displays** panel, click **Security Editor**.
3. In the Members area, select the user you want to view or edit, and then click **Edit/View Member**.
The **Security Editor** displays the information for the selected user.
4. If applicable, make the required changes to the user.
5. Click **Save**.

Related information

[Edit or View a Group for a User \(on page 344\)](#)

[Add workstations to key sets for a group \(on page 374\)](#)

[Add key sets to a workstation for a group \(on page 375\)](#)

Workflow Group Details

The Workflow group **Details** tab on the **Security Editor** for Workflow groups displays all of the key set information for the selected group.

Edit Description

You can change the description of this resource.

Attribute	Value	Description
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

Key Sets

The following table describes the information available in this section:

Attribute	Value	Description
Display Options	User-defined.	Select the method you want to use to display the key sets assigned to the group. You can display by key set, by resource, or by workstation.
Resource	User-defined	Displays all of the resources that the selected group holds key sets for. This column is displayed when you select either the Show by Key Set or Show by Workstation option.
Workstation	User-defined	Displays all of the individual workstations, including All Workstations if applicable, that the selected group holds key sets for. This column is displayed when you select either the Show by Key Set or Show by Resource option.
Group	User-defined	Displays the group that holds key sets for the resource. This column is displayed when you select either the Show by Resource or Show by Workstation option.

Attribute	Value	Description
Key Set	User-defined	Displays the key sets that the selected group holds. This column displays when you select either the Show by Resource or Show by Workstation option.
Assign Key Sets	N/A	Click to display the system keys in the Security Editor . You can assign or remove key sets for the selected group or user.
Edit/View Resource	N/A	Click to display the selected resource's information in the Security Editor . Select a resource or key set (depending on the option selected) to enable this link. You can view or edit the resource's information.

Workflow Group Members

The **Workflow Users** tab on the **Security Editor** for Workflow groups displays the list of Workflow users who are members of the selected group. You can add additional members to the group or remove members.

Workflow Users

The following table describes the information available in this section:

Attribute	Value	Description
Members List	User-defined	Displays the users who are members of the current group.
Edit/View Member	N/A	Click to display the selected user's information in the Security Editor . You can view or edit the user's information.

Attribute	Value	Description
Add/Remove Members	N/A	Click to open the Add/Remove Members dialog box.
Add/Remove Members : available members list	User-defined	Displays the users that are available to add to the selected group.
Add/Remove Members : members list	User-defined	Displays the list of users that are already members of the selected group.

Domain-Based Security

Windows Domain-Based Security

You can use Windows domain user accounts to manage access to and within Workflow. You can also configure domain user accounts for automatic login on either a global basis or for individual computers.

By using Windows domain user accounts, you gain the following advantages of Windows security in relation to Workflow login:

- Case sensitive passwords
- Passwords that expire
- Online password changes
- Ability to specify minimum password requirements
- Account lockout

Windows domain user accounts also provide options for controlling access to Workflow functionality after users have logged in to the system. Specifically, you can enforce security based on the permissions afforded by either:

- Workflow groups that you directly associate with users one at a time, and/or
- Workflow groups that you map to active directory universal or global groups and, thus, to all related users at once.



Important:

Only Windows domain user accounts are supported, and you cannot remove user accounts from Windows domains from within Workflow.

Login Security Management

Workflow requires that any Windows domain user be authenticated by active directory. To designate a Windows domain user in Workflow, you must name the corresponding Personnel resource in the format of `<Fully Qualified Windows Domain Name>\<Windows Account Name>` (for example) `MyDomain.MyCompany.com\RobertGray`). In response, the system automatically creates a Workflow user account that provides for access to the system based on the user's Windows credentials.



Note:

Beginning with Workflow 2.0, you can add non-Windows domain users to the Windows Users group in the Personnel model. However, Workflow does not consider those users to be Windows domain users for security purposes; rather, they are considered Workflow users who are members of a Workflow group named Windows Users. You must manually create accounts and specify system access credentials for such users, and cannot control their access to Workflow functionality by way of active directory mapping.

In attempting to log in a Windows domain user, Workflow sends the user's Windows user name and password to a Windows domain controller for authentication. If the controller verifies the user name and password, Workflow completes the login process; otherwise, an error message appears alerting you to the failure.



Note:

To log in to Workflow, the system must be in a state of either Complete or Partial health. For more information, see **System Health and Program Use**. If autologin is configured for Windows domain users—either for the current computer or all computers in the system—Workflow will log in with the currently logged in Windows domain user.

Following initial log in, Windows domain users are automatically designated as such for the purpose of system access, as reflected on the **Domain Users** tab of the **Security Editor**.

Post-Login Security Management

In relation to Windows domain users, you can control access to Workflow functionality based on either the Workflow groups that you directly associate with the user, or the active directory groups that the user belongs to and that you, in turn, map to Workflow groups. Users inherit the key sets and corresponding permissions defined for the Workflow groups that they are associated with either directly or by way of active directory group mapping.

**Note:**

These two approaches are mutually exclusive for any given Windows domain.

Direct group association and active directory mapping both offer the benefit of centralized, Windows domain-based credential management for access to the overall system. In deciding which approach to adopt for controlling access to Workflow functionality, consider the following aspects of each:

- Directly associating Windows domain users with Workflow groups provides for a comparatively greater degree of flexibility in granting access to the Workflow functionality, because any such user can be provided broader or narrower permissions than other users in the same Windows domain active directory group. However, maintenance efforts may be greater, because the security configurations must be managed on a per-user basis.
- Mapping active directory groups to Workflow groups provides for a comparatively lesser degree of flexibility in granting Windows domain user access to the Workflow functionality, because all users in an active directory group are subject to the same security constraints within the system. However, an active directory group can be mapped to many Workflow groups, and a Workflow group can have many active directory groups mapped to it. Furthermore, changing the mapping between active directory groups and Workflow groups changes the permissions that are granted to all Windows domain users who are members of the mapped active directory groups at once, which can speed maintenance efforts. Additional time savings can also be realized when adding new Windows domain users who are members of previously mapped active directory groups, as these users are automatically and immediately afforded the permissions provided by the related Workflow groups.

**Note:**

Windows domain users that were created in versions of Workflow prior to 2.0 will continue to function as before assuming that the active directory groups they are members of are not mapped to any Workflow groups.

User Type Conversion

Beginning with Workflow version 2.2, you can convert Workflow users to domain users, and vice versa, by either adding the applicable domain designation to the user's name or removing the designation from the name, respectively.

If you convert a Windows domain user to a Workflow user, you must subsequently specify access credentials for the user, and, if the permissions that were previously afforded the user, which carry over with the conversion, are no longer appropriate, adjust group associations.

Windows Domain Users Logging into Workflow

If you are using Windows user names and passwords within Workflow security, be aware that Windows user accounts must have the policy *Access this computer from the network* applied under the local security policy. By default, this policy is assigned to the groups *Users* and *Everyone* on the local machine.

If the domain policy overrides the local policy settings by removing these groups, then the Windows user names and passwords will fail with insufficient rights when trying to log in to Workflow. If domain administrators want to restrict this right, then they must do one of the following tasks in order to continue to use Windows user names and passwords within Workflow:

- Create a Domain Group that contains all the Domain Users that will be used within Workflow security, add this group to the domain policy *Access this computer from the network*, and deploy this policy to all machines running Workflow.
- Add the Domain Users group to the domain policy *Access this computer from the network*, and then deploy this policy to all machines running Workflow.
- Add Authenticated Users to the domain policy *Access this computer from the network*, and then deploy this policy to all machines running Workflow. Be aware that this group requires each user to log on to the domain at least once to be considered an authenticated user.
- Leave at least the Users group in the domain policy *Access this computer from the network*. If you choose this option, be aware that the Anonymous user and the Guest user are not part of the Users group.

Updating Existing Windows Domain Users in Workflow

If you have existing Windows domain users that were created in a previous version of Workflow, and those user accounts are now associated with one or more active directory universal or global groups, the first time those users log in, whether manually or automatically, the system will update the domain information for each user and automatically remove those users from the pre-existing Windows Users group in Workflow.

Related information

[Automatic Login Authentication \(on page 389\)](#)

Authenticate Windows Domain Users for Vision Calls

Vision uses a token to make a secure call. For Windows authentication, the Proficy STS service needs to query the domain controller to retrieve groups that the user belongs to. This group information is used to validate the Windows user.

Troubleshooting Windows Authentication

By default, the local system account is used to run the Proficy STS service. In most cases, the local system account can retrieve user group information by querying the domain controller. However, your environment may restrict this local system account from performing this query against the domain controller. In this case, the Windows domain user cannot be validated and no security token can be issued, thereby rejecting the Vision call.

To solve this problem, run the Proficy STS service under a domain user account. This enables this service to query the domain controller.

**Note:**

If this solution fails, assign the domain account to be a member of the local administrator group, and then try again.

Related information

[Security Authorization \(on page 336\)](#)

Password Expiration Considerations

When Workflow security is synchronized with Windows security, passwords can expire.

If the Windows password has expired, the user is notified and prompted to change the password. If the Windows password is about to expire, a notification message displays, reminding the user to change the password.

Configuring Domain-Based Access Control to Workflow Functionality

You can use active directory universal or global groups to control access to Workflow functionality.

By mapping an active directory group to a Workflow group, you give the key sets and corresponding permissions defined for the Workflow group to all members of the active directory group at the same time. An active directory group can be mapped to many Workflow groups, and a Workflow group can

have many active directory groups mapped to it. The operator must have permission to query Windows domains.



Note:

For Vision, when using the Predix token (Windows user) to access the Workflow account that runs the Proficy Server service, an operator must have permission to query the domain(s); otherwise, authentication will fail as no group information can be retrieved from the domain.

For more information about universal groups, see [http://technet.microsoft.com/en-us/library/cc755692\(v=ws.10\)](http://technet.microsoft.com/en-us/library/cc755692(v=ws.10)).

Assigning operating system rights to Windows domain user accounts

When you use Windows security in Workflow, user accounts that need to log in to a machine must have the *Act as Part of the Operating System* right enabled in the local security policy.

Before you begin



Note:

User accounts that are not used to log in to Windows should not have this right.

Procedure

1. From the **Start** menu, click **Control Panel > Administrative Tools**.
2. Double-click **Local Security Policy**.
The Local Security Policy dialog box appears.
3. Expand the **Local Policies** folder.
4. Select the **User Rights Assignment** folder.
5. In the **Rights** list, double-click **Act as Part of the Operating System**.
The **Act as part of the operating system Properties** dialog box appears.
6. Add the users that you want to have this right, and then click **OK**.
7. Close the remaining dialog boxes, log out of Windows, and then log in again for your changes to take effect.

Set Windows passwords to expire

One of the benefits of using Workflow with Windows security is that you can set Windows passwords to expire.

Procedure

1. From the **Start** menu, click **Control Panel > Administrative Tools**.
2. Double-click **Local Security Policy**.
The Local Security Policy dialog box appears.
3. Expand the **Account Policies** folder.
4. Select **Password Policy**.
The password policies display in the **Policy** list.
5. Double-click **Maximum password age**.
The **Maximum password age Properties** dialog box appears.
6. In the **Password will expire in** field, set the number of days after which passwords expire.
7. Click **OK**.

Limit the number of invalid login attempts

When a Workflow user is connected to a Windows domain user account, you can set an account lockout threshold, which prevents a user from accessing the account after he enters the incorrect user name or password beyond the number of acceptable times. When the account lockout threshold has been reached, the account is disabled.

Procedure

1. From the **Start** menu, click **Control Panel > Administrative Tools**.
2. Double-click **Local Security Policy**.
The Local Security Policy dialog box appears.
3. Expand the **Account Policies** folder.
4. Select the **Account Lockout Policy** folder.
The account lockout policies display in the **Policy** list.
5. Double-click **Account Lockout Threshold**.
The **Account lockout threshold Properties** dialog box appears.
6. In the **Account will not lock out** field, enter the number of invalid login attempts before the account is disabled.
7. Click **OK**.

Associate Active Directory universal or global groups with Workflow groups

Domain users are Windows users who are members of one or more Windows domain groups.

About this task**Important:**

You must have created your Windows users and added them to the appropriate active directory universal or global groups before you can map these groups to Workflow groups. However, a domain user may join a domain group that has already been mapped to a Workflow group.

Procedure

1. In the navigator, click **Personnel > Groups**.
2. From the list of groups, select the group you want to associate an active directory group with.
3. In the **Displays** panel, click **Security Editor**.
4. Click the **Domain Users** tab, and then use one of the following methods to map active directory groups to the Workflow group.

Method	Description
<p style="text-align: center;">Browse for unmapped active directory universal or global groups</p>	<ol style="list-style-type: none"> a. Click Add/Remove Groups. <p style="margin-left: 20px;">The Mapping Groups dialog box appears.</p> b. From the Available Trusted Domains list, select the domain containing the active directory groups that you want to map to the selected Workflow group. c. To find unmapped groups on the selected domain, in Search for universal or global groups, enter search criteria. <div data-bbox="938 1333 1421 1600" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note: The maximum number of results supported is 50. If more than this number is returned, you must narrow your search criteria.</p> </div> d. From the Unmapped Domain Groups list, select the active directory groups you want to map, click the right arrow, and then click OK.

Method	Description
	<div data-bbox="862 258 1422 1346" style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px;"> <p> Tip:</p> <ul style="list-style-type: none"> ◦ You can enter search criteria using a wildcard character (*) combined with other characters. ◦ To add multiple active directory groups to a Workflow group, press the Ctrl key, and then select all of the active directory groups that you want to add to the Workflow group, or click the double right arrow to add all of the active directory groups. ◦ To remove an active directory group association from a Workflow group, from the Mapped Domain Groups list, select the active directory group that you want to remove, and then click the left arrow. Press the Ctrl key to select multiple groups, or click the double left arrow to remove all of the mapped active directory groups. </div>
<p>Edit the active directory universal or global groups list</p>	<ol style="list-style-type: none"> a. In the Domain Name column, double-click in the first available row. b. Enter the name of the domain that contains the active directory groups that you want to map to the current Workflow group. c. In the Group Name column, double-click the cell, and then enter the name of the active directory group you want to map.

Method	Description
Copy and paste a list of active directory groups from an Excel spreadsheet	<ol style="list-style-type: none">a. Open the Excel spreadsheet that contains your domain names and active directory group names.b. Copy the two columns of data (the first column contains the domain names; the second column contains the active directory group names).c. In Workflow, on the Domain Users tab, in the Active Directory groups table, double-click any cell, and then paste the data from the Excel spreadsheet. <div data-bbox="943 793 1422 1056"><p> Note: If there is existing data in the table, then the newly pasted data is added after the last row of the existing data.</p></div> <div data-bbox="862 1087 1422 1434"><p> Tip: You can also copy the data from the Active Directory groups table to an Excel spreadsheet. Press the Ctrl key to select multiple rows to copy, then in Excel, select a single cell, and then paste the data.</p></div>

5. Click **Save**.

The active directory groups are mapped to the selected Workflow group and are displayed in the **Active Directory groups** list. The domain users who are members of the mapped active directory groups are displayed in the **Domain Users** list.

**Note:**

All domain names are validated against the domain before they are saved in Workflow. If the domain or active directory group name is entered incorrectly, an error message appears and the row is cleared.

Automatically create Workflow user accounts for Windows domain user accounts

You can automatically create a Workflow user when a Windows domain user logs into Workflow client for the first time.

Before you begin

A Windows domain user's parent group must have been mapped to a Workflow group before the user can log in to Workflow client.

Procedure

Instruct a new Windows domain user to log in to Workflow client.

**Important:**

This must be the first login attempt by the Windows domain user.

A new Workflow user account is created for the Windows domain user.

Manually add a Windows domain user to Workflow

You can use Windows domain user accounts to manage access to and within Workflow.

Before you begin

A Windows user's parent group must have been mapped to a Workflow group before the user can log in to Workflow client.

Procedure

1. In the navigator, click **Personnel > People**.
2. Click **Add User**.

3. In the **Name** field, enter the name for the user in the format `<Fully Qualified Windows Domain Name>\<Windows Account Name>` (for example `MyDomain.MyCompany.com\RobertGray`).



Note:

Because the Windows account name is not verified against the domain when creating the user, be sure to enter the correct account name; otherwise, the user will not be able to log in to Workflow.

4. In the **Description** field, enter clarifying information about the user.
5. Click **OK**.

Convert between Windows domain and Workflow users

Beginning with Workflow version 2.2, you can convert Workflow users to domain users, and vice versa, by either adding the applicable domain designation to the user's name or removing the designation from the name, respectively.

Prerequisites

[Manually add a Windows domain user to Workflow \(on page 367\)](#)

[Add Users and Personnel Classes \(on page 340\)](#)

Procedure

1. In the navigator, click **Personnel > People**.
2. In the **Folders** list, click the user that you want to convert the type for.
3. In the **Displays** panel, select **Production Management > S95 Model Editor**.
4. On the **Details** tab, click the **Edit** link.
5. Edit the **Name** value as appropriate to the conversion direction:
 - If you are converting a Workflow user to a Windows domain user, edit the value to reflect the format `<Fully Qualified Windows Domain Name>\<Windows Account Name>` (for example `MyDomain.MyCompany.com\RobertGray`).
 - If you are converting a Windows domain user to a Workflow user, remove the domain designation and backslash from the value.
6. Click **Save**.

What to do next

If you convert a Windows domain user to a Workflow user, you must subsequently specify access credentials for the user, and, if the permissions that were previously afforded the user, which carry over with the conversion, are no longer appropriate, adjust group associations.

Related information

[Configure Security Credentials for a User \(on page 342\)](#)

[Add One or More Users to a Group \(on page 353\)](#)

Active Directory Groups and Domain Users

The **Domain Users** tab displays the list of active directory universal or global groups and Windows domain users who are members of those groups. You can map additional active directory groups to the current Workflow group or remove mapped active directory groups.

Domain Users

Domain users are Windows users who are members of one or more Windows domain groups.

The following table describes the information available in this section:

Attribute	Value	Description
Domain Users List	User-defined	Displays the domain users who are members of active directory universal or global groups that have been mapped to the current Workflow group.
View User	N/A	Click to display the selected user's information in the Security Editor .

Active Directory Groups

The following table describes the information available in this section:

Attribute	Value	Description
Domain Name	User-defined	Displays the name of the domain whose active directory groups are mapped to the current Workflow group.

Attribute	Value	Description
Group Name	User-defined	Displays the name of the active directory group that is mapped to the current Workflow group.
Add/Remove Groups	N/A	Click to open the Mapping Groups dialog box.

Mapping Groups Dialog Box

The following table describes the information available in this section:

Attribute	Value	Description
Mapping Groups: Available Trust Domains	User-defined	Displays the list of available domains that can be mapped to the current Workflow group.
Mapping Groups: Unmapped Domain Groups	User-defined	Displays the list of active directory groups that belong to the selected domain and that have not yet been mapped to Workflow groups.
Mapping Groups: Mapped Domain Groups	User-defined	Displays the list of active directory groups that are already mapped to the current Workflow group.
Search for universal or global groups	User-defined	Search to find unmapped groups on the selected domain. You can search using a wildcard character (*). <div data-bbox="1024 1570 1427 1810" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: The maximum number of results supported is 50. If more than this number is returned, you must </div>

Attribute	Value	Description
		 narrow your search criteria.

Key Sets

Key sets are groups of permissions, based on areas of operation within your facility, that define the operations a group of users can perform.

After you have configured your key sets, you assign the key sets to the appropriate groups. If a user is a member of a group that has been assigned a specific key set, the user has all of the permissions configured for that key set.

Workflow provides default key sets that you can assign permissions to. These key sets are:

- System Visibility
- S95 Configurator
- Workflow Author



Note:

When you install Workflow, you are required to create an administrator user. The administrator user is automatically assigned all permissions for all key sets.

The default key sets are hierarchical with System Visibility, S95 Configurator, and Workflow Author visible at the top. S95 Configurator and Workflow Author contain default child key sets. You can add key sets to any level of the hierarchy below the top level key sets. The following table describes the hierarchy of the default key sets.

> System Visibility				
> S95 Configurator	> Equipment Configurator > Material Configurator > Personnel Configurator			

Workflow Author | > Workflow Builder | > Workflow Manager | > Workflow Supervisor | > Workflow Operator

**Note:**

The Administrator key set is at the top of the hierarchy, but because it has all permissions and cannot be modified, it is not visible.

The following table describes the top level key sets.

Key Set	Description
System Visibility	Required for any user to see the models list.
S95 Configurator	Automatically assigned permissions to create, edit, and delete resources in the Equipment, Material, Personnel models.
S95 Configurator > Equipment Configurator	Automatically assigned permissions to create, edit, and delete resources in the Equipment model.
S95 Configurator > Material Configurator	Automatically assigned permissions to create, edit, and delete resources in the Material model.
S95 Configurator > Personnel Configurator	Automatically assigned permissions to create, edit, and delete resources in the Personnel model.
Workflow Author	Automatically assigned permissions to create workflows and subprocesses to be used by workflow builders.
Workflow Author > Workflow Builder	Automatically assigned permissions to create workflows by assembling predefined subprocesses.
Workflow Author > Workflow Manager	Automatically assigned permissions to maintain workflows and historical task data in the system by importing and exporting data, and by archiving and purging data.
Workflow Author > Workflow Supervisor	Automatically assigned permissions to handle task assignment and management for their assigned operators.

Key Set	Description
Workflow Author > Workflow Operator	Automatically assigned permissions to perform work based on tasks that are assigned and available within a Task List.

You can create as many key sets as you require and can organize them into a hierarchy that fits your business purposes. You can also assign additional permissions to each of the default key sets. You can delete the default key sets and replace them with your own; however, you cannot rename them. If a key set is already in use in your system, you cannot delete it.

The permissions that you assign to each key set are inherited by its parent. For example, you have the following key sets: Workflow Operator, Workflow Supervisor, and Workflow Manager. If you assign the system permission *Visible* and the workflow permission *Cancel Tasks* to the Workflow Operator key set, then the parent key set, Workflow Supervisor, inherits those permissions. If you then assign the *Execute Workflows* and *Execute Schedules* permissions to the Workflow Supervisor key set, those permissions, as well as the *Visible* and *Cancel Tasks* permissions are inherited by the Workflow Manager key set.

In addition, permissions set at one level are not inherited by child key sets. Therefore, the Workflow Supervisor key set does not automatically inherit the permissions assigned to the Workflow Manager key set.

Because the Administrator key set automatically has all permissions, it is not available when you are assigning key sets to groups.

When you are designing the security plan for your facility, make sure that you take key set assignments into consideration. When you are assigning key sets, you assign them to the appropriate solution and displays (found under Solutions in the Proficy System model).



Note:

You can also assign key sets for the System using the **Assign Key Sets Quick Find** link under the Personnel model in the navigator or by using the **Assign Key Sets** link in the **Security Editor** for a specific user or group.

Assign permissions to a key set

Use the **Security Editor** to assign permissions to key sets.

Procedure

1. In the navigator, click **Personnel > Key Sets**.
2. Select the key set that you want to assign permissions to.
3. In the **Key Sets** list, select a key set to configure.
4. In the **Displays** panel, click **Security Editor**.
The table listing all permissions appears.
5. In the **Permissions** table, in the **Allow** column, select the check box for each permission you want to assign to a key set.
6. Click **Save**.

Edit or view a resource for a key set

Use the **Security Editor** to view or edit key sets that a user has inherited from the parent group(s).

Before you begin

Individual users cannot hold key sets. A user must be a member of one or more groups and will inherit the key sets assigned to the group.

Procedure

1. In the navigator, click **Personnel > People**.
2. Select the user whose key sets you want to view or edit.
3. In the **Displays** panel, click **Security Editor**.
4. In the **Key Sets** area, select the key set with the resource you want to view or edit, and then click **Edit/View Resource**.
The **Security Editor** displays the information for the selected resource.
5. If applicable, make the required changes to the resource.
6. Click **Save**.

Add workstations to key sets for a group

This procedure guides you through the task of adding workstations to keys sets at the global, equipment resource, and solution display levels. Only the access path is different.

Before you begin

You must have created your resources before assigning key sets.



Tip:

You can also assign key sets globally by clicking **Assign Key Sets** in the **Quick Find** list for the Personnel model.

Procedure

1. Access the Security Editor.
2. In the **Select a Group** area, select the group you want to assign key sets to.
3. In the **Permission Details For** area, select the **Edit by Key Set** option.
4. In the **Edit by Key Set** area, under the key set you want to add a workstation to:

Click...	To...
Add Workstation	Add a specific workstation to the key set for the selected group.
Add All Workstations	Add all workstations in the system to the key set for the selected group.

**Tip:**

You can remove a workstation from a key set by selecting the workstation you want to remove, and then clicking **Remove Workstation**. However, you cannot remove a workstation from the group if you assigned it directly to the group through the Personnel model.

5. **Optional:** Repeat step 4 for each key set you want to add a workstation to for this group.
6. **Optional:** Repeat steps 3-5 for each group you want to add workstations and key sets to.
7. **Optional:** Repeat steps 1-6 for each solution display you want to add workstations and key sets to.
8. When you are finished adding workstations, click **Save**.

Add key sets to a workstation for a group

This procedure guides you through the task of adding key sets to workstations at the global, equipment resource, and solution display levels. Only the access path is different.

Procedure

1. Access the Security Editor.
2. In the **Select a Group** area, select the group you want to assign key sets to.
3. Select the **Edit by Workstation** option.
4. In the **Edit by Workstation** area, under the workstation you want to assign a key set to, click **Add Key Set**.

**Tip:**

You can also add key sets for all workstations.

The **Universal Browser** appears.

5. Select the key set you want to add to the workstation for the selected group, and then click **OK**.



Tip:

You can remove a key set from a workstation by selecting the key set you want to remove, and then clicking **Remove Key Set**. However, you cannot remove a key set from the group if you assigned it directly to the group through the Personnel model.

6. **Optional:** Repeat step 4 for each key set you want to add to a workstation for this group.
7. **Optional:** Repeat steps 3-6 for each group and workstation you want to add key sets to.
8. When you are finished adding key sets, click **Save**.

View all key set permissions for a group

Before you begin



Note:

The assigned key sets and workstations for the selected group are different depending on whether you are at the system or solution display level.

Procedure

1. Access the **Security Editor**.
2. From the **Select a Group** list, select the group you want to view permissions for.
3. Select the **Show all Permissions** option.

All of the permissions for the group are displayed with the associated key sets and workstations.

Key Set Assignment Examples

Workflow security is location-based; that is, you assign permissions to a personnel group based on equipment locations and workstations within your facility. Security is a combination of permissions (key sets), plus workstation(s), plus location, and this combination is assigned to a group.

You can assign key sets for a group on individual pieces of equipment, and by workstation. You can also assign key sets on a global basis.

A key set is a collection of permissions. If you assign a key set to a group for a specific location, that group can perform all the tasks allowed in that key set. You cannot assign key sets or add workstations for individual users.

A personnel group is a collection of users with the same access rights to your facility's processes. When you assign key sets to a group, each member of the group inherits the permissions defined for the key set.

Example 1

You have a group called *Line 1 Supervisors* and you have defined the *Create Equipment* permission for the **Equipment Configurator** key set. You have assigned the *Line 1 Supervisors* group to Line 1 in your facility with the **Equipment Configurator** key set and *Workstation D*. This configuration allows the *Line 1 Supervisors* group to create equipment resources for *Line 1*, but only when logged on to *Workstation D*. If you want the *Line 1 Supervisors* to be able to create equipment resources for *Line 1* from any workstation, you can add all workstations to the **Equipment Configurator** key set for this group.

Globally Defined Key Sets

If you want the *Line 1 Supervisors* to be able to create equipment resources for all locations in your facility, you can configure this at the global level. Be aware that if you configure this at the global level, you cannot remove the key set from an individual location. To configure this group to be able to create equipment resources on specific locations, you must configure the key set and workstation for the group on each individual location.

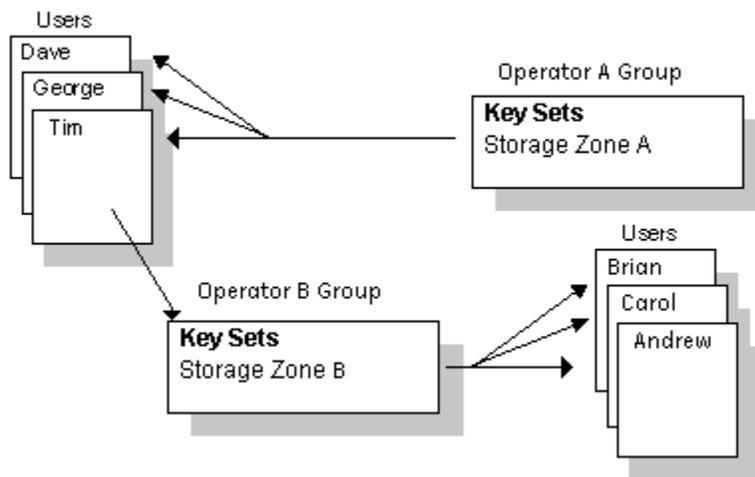
When you configure security for a selected group at the global level, those assignments are visible at all locations. For example, if you assign the **S95 Configurator** key set for a single workstation at the global level, that key set and workstation combination is defined for the selected group on all equipment locations in your enterprise. You cannot remove this key set/workstation from an individual location; you can remove or change it only at the global level.

Solution Display Key Sets

In addition to assigning key sets to a group and location, you can assign key sets to a group for a solution display. When you assign a key set with the **Visible** permission to a solution display for a group, that display is visible for the members of that group at all times, regardless of the workstation the user is logged on to. For example, if you want to make the **Report View** visible only to the Administrators group, set the **Visible** permission on the **Administrators** key set and then add that key set to the Administrators group for the **Report View** solution display.

Example 2

You have three users, Dave, George, and Tim, who are all members of the Operator A group. All three users need access to Storage Zone A, while Tim needs additional access to Storage Zone B. In order for Tim to have access to Storage Zone B, he must be a member of the group Operator B, which has the key set for Storage Zone B assigned to it.



Related information

[Workflow Groups \(on page 352\)](#)

Key Set Permissions (Security Editor)

The **Security Editor** provides the ability to configure the permissions for your key sets, as well as for the default key sets.

You can change the description of this resource.

Attribute	Value	Description
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

Permissions

The following lists describe the permissions you can assign when configuring your key sets. You can allow access to:

- System resources to:
 - view and access resources
- Database to:
 - execute database SQL queries/nonqueries
 - create and manage database connections

- create and manage database statements
- import and export database connections and statements
- Email to:
 - configure server-wide emails for sending messages
 - send email messages
- Equipment resources to:
 - create, edit, and delete elements of the equipment instances and equipment classes, and to modify their properties
 - read elements of the equipment hierarchy and equipment classes
 - import and export equipment resources
- Event resources to:
 - configure events
 - enable and disable events
 - import and export events
 - archive and purge events
- File system resources to:
 - create and delete file watches
- Form and display resources to:
 - create and edit forms and displays
 - delete forms and displays
- Historian to:
 - access a specific Historian server's data

**Note:**

This permission is provided for use by Historian Analysis.

- IT-related resources to:
 - configure computers and data sources
 - import and export computer configurations
- Material resources to:
 - create, edit, and delete material resources and material classes
 - import and export material resources
 - archive and purge material lots
- Personnel resources to:
 - create, edit, and delete people (users) and personnel classes
 - configure user credentials, including changing user names and passwords
 - configure groups and key sets

- add users to and remove users from groups
- assign key sets to users and groups
- import and export personnel resources
- Production resources to:
 - create, edit, and delete production resources
 - import and export production resources
 - archive and purge production resources
- Projects to:
 - create, edit, and delete project definitions
 - package projects
 - deploy projects
- Property group resources to:
 - edit property groups
- Regulatory compliance records and configuration to:
 - archive and purge audit and electronic signature records
 - configure audit and electronic signature behavior
- Reporting schedule editor to:
 - configure reporting schedules
 - manually start data synchronization
- Vision to:
 - create users and manage authorization
 - create, edit, and delete S95 equipment model data and properties
 - create, edit, and delete shared favorite folders
 - reset passwords, manage all users, change system settings, and assign the ManageAuthorization and ManageSystem permissions
 - view S95 equipment model data and properties
 - view shared favorite folders
- Web Service to:
 - execute web service methods
 - import and export web services
 - create and manage web services
- Workflow resources to:
 - administer tasks, including setting task priority, re-assigning tasks, and releasing tasks
 - archive and purge historical task data
 - cancel tasks and task steps
 - configure workflows, subprocesses, categories, schedules, user activities, and My Task List Editor

- debug workflows
- execute workflows and schedules
- import and export workflow assets
- jump over task steps
- perform workflow tasks, including entering data, starting tasks manually, acquiring tasks, releasing tasks, and entering expiration comments
- configure personal and predefined filters for users in the Task List

For appropriate key sets, you can also select the **Select All Operations** check box, which gives users with that key set access to all areas of your facility.

Related information

[Workflow Groups \(on page 352\)](#)

[Workflow Users and Personnel Classes \(on page 339\)](#)

Key Set and Workstation Assignments

Workflow security is location-based; that is, you assign permissions to a personnel group based on equipment locations and workstations within your facility. Security is a combination of permissions (key sets) plus workstation(s) plus location, and this combination is assigned to a group.

Assigning Key Sets and Workstations

You have the option of adding workstations to key sets or adding key sets to workstations. You may find that one way works better for you than the other. For example, if you have many workstations, but only a few key sets, you may prefer to add workstations to key sets. If so, use the **Edit by Key Set** option. If you have many key sets, but only a few workstations, use the **Edit by Workstation** option.

The following table describes the common information and options available for security configuration.

Property	Value	Description
Groups	User and system-defined	Displays a list of all of the personnel groups defined for your system. This list includes the default groups Administrators, Guests, and Windows Users.

Property	Value	Description
Members of	User-defined	Displays a list of all of the users who are members of the selected group.
Edit by Key Set	N/A	Select this option to add workstations to key sets for the selected group.
Edit by Workstation	N/A	Select this option to add key sets to workstations for the selected group.
Show all Permissions	N/A	Select this option to display all of the permissions, key sets, and workstations associated with the selected group.

Edit by Key Set

In this view, you can add individual workstations to key sets, or add all workstations to specific key sets. The following table describes the information and options specific to the **Edit by Key Set** option.

Property	Value	Description
Key Sets	User and system-defined	Displays all of the key sets defined for your system. This list includes the default key sets and the key sets that you created.
Workstation(s)	User-defined	Displays all of the workstations assigned to the key set for the selected group and location.
Remove Workstation	N/A	Click to remove the selected workstation from the key set. This link is available only when the Edit by Key Set option is selected.
Add Workstation	N/A	Click to open the Universal Browser to select a single work-

Property	Value	Description
		station to add to the key set for the selected group and location. This link is available only when the Edit by Key Set option is selected.
Add All Workstations	N/A	Click to add all of the workstations in your system to the key set for the selected group and location. This link is available only when the Edit by Key Set option is selected.

Edit by Workstation

In this view, you can add individual key sets to workstations. The following table describes the information and options specific to the **Edit by Workstation** option.

Property	Value	Description
All Workstations	N/A	Represents all of the workstations defined for your system.
Workstation(s)	User-defined	Displays the individual workstations defined for your system.
Key Set(s)	User and system-defined	Displays the key set(s) assigned to the workstation(s) for the selected group and location.
Remove Key Set	N/A	Click to remove the selected key set from the workstation. This link is available only when the Edit by Workstation option is selected.
Add Key Set	N/A	Click to open the Universal Browser to select a key set to add to the workstation for the selected group and location.

Show All Permissions

At any time, you can view all of the permissions, key sets, and workstations configured for a group by selecting the **Show all Permissions** option. To view all key sets and workstations configured for a group at both levels, access the group resource in the Personnel model.

Related reference

[Key Set Permissions \(Security Editor\) \(on page 378\)](#)

Related information

[Workflow Users and Groups \(on page 338\)](#)

[Workflow Security \(on page 331\)](#)

[Workflow Groups \(on page 352\)](#)

Login and Logout Capabilities

Workflow access can be configured in multiple ways. You can configure automatic login and logout on individual computers, on all computers in your system, or on a subset of computers in your system. You can also retain manual login and logout on a global basis or on specific computers.

By logging in, users identify themselves as Workflow users or Windows domain users and gain access to resources that they are authorized to use. Automatic login can be configured for a specific Workflow user, or the currently logged in Windows domain user on individual workstations, or can be configured to automatically log in the currently logged in Windows domain user on either individual workstations or on all computers in the system.

User Type	Autologin to Individual Computers?	Autologin to All Computers?
Workflow user	Yes	No
Windows user	Yes	Yes

Manual login means that, regardless of permissions and key set configuration for users and groups, login credentials must be entered manually to access Workflow.



Note:

To log in to Workflow, the system must be in a state of either Complete or Partial health. For more information, see **System Health and Program Use**.

Automatic Login and Logout

Automatic login and logout can be configured for individual workstations or for all workstations that are included in a global default configuration. You can configure automatic login for Windows users on individual computers or on a global basis. You can configure automatic login for Workflow users only on individual computers.

Workflow checks for changes in configuration settings during the login and logout procedures and posts an information message in the Feedback Zone every time it detects changes.

Automatic Login

Automatic login allows you to specify the user account that will be automatically logged in when Workflow is opened on a specific workstation. You can change the configuration of the default account and enable or disable automatic login.

**Note:**

By default, Workflow is configured with automatic login disabled.

You can configure a default user with specific permissions, and when Workflow starts, if a default layout is configured for the default account, it is displayed.

When automatic login is enabled on a workstation, the client runs under the default user account until a user logs in with his or her own account credentials. When the user logs out, Workflow automatically logs back in and runs under the default account.

**Note:**

Automatic login does not take effect on remote clients until a user has manually logged in once.

If automatic login is disabled, Workflow runs under the credentials that the user entered through the login dialog box.

If the default user credentials are modified, Workflow logs in with the old credentials, detects the changes and then logs out and logs in again using the new credentials.

If you are configuring global default computer settings, you can also configure automatic login. Automatic login for global configurations is available only for Windows user accounts. If automatic login is enabled, then when the currently logged on Windows user opens Workflow, he will be automatically logged in based on his authenticated login credentials. If the authentication fails, the **Workflow Login** dialog box will appear, and the user will be required to enter his login credentials manually.

**Note:**

If changes are made to the automatic login configuration, they do not take effect immediately. You must log out and then log back in for the changes to take effect.

User Login

When a user logs in manually, the Workflow client operates under the user account. If automatic logout is enabled, Workflow reverts back to the default user account when the user logs out.

If automatic login is disabled, the client displays the login dialog box when the user logs out.

Automatic Logout

You can configure the amount of time that Workflow remains active without any user activity before the user is logged out of the system.

When automatic logout is enabled, Workflow logs out the current user after a specified time of inactivity and remains logged out if automatic login is disabled.

If a computer is configured for automatic login and automatic logout, and is logged in with the automatic login credentials but is inactive for the specified amount of time, Workflow will automatically log out and then log back in with the automatic login credentials. However, if another user logs in using different credentials and is inactive for the specified amount of time, Workflow will automatically log that user out, and then log in with the defined automatic login credentials.

You can also configure automatic logout if you are configuring global default computer settings.

This feature does not eliminate the need to manually log out when an operator finishes using Workflow, particularly if you have strict security requirements. If you decide to use this feature, consider it as a safety mechanism that prevents operators from remaining logged in indefinitely.

Automatic Login and Logout Behavior Examples

These examples describe how Workflow behaves during specific autologin and autologout situations.

Scenario 1

Automatic login is configured and a user expects automatic login to succeed on the first attempt to log in to a remote client.

Result

The **Workflow Login** dialog box appears.

Enter valid user credentials, and then click **OK**.

Scenario 2

Changes are made to the automatic login credentials.

Result

After logging out, attempts to automatically log in using the **Autologin** button fail.

Click **Cancel** on the **Workflow Login** dialog box, click the **Log in to Proficy** icon, and then click **Autologin**.

When the client is restarted, automatic login will succeed with the configured credentials.

Scenario 3

The application server is stable during Workflow client startup, but the configured credentials for autologin are invalid. A message displays stating, "The user credentials configured for Autologin are invalid. Please ask Administrator to check the configured credentials."



Note:

This behavior applies to both Workflow user and Windows user authentication.

Result

Click **OK**. The Workflow client changes to logged out mode displaying the default login view. You can now click the **Login** button and log in as a primary user.

Scenario 4

You cancel the login dialog box in autologin mode.

Result

The Workflow client logs out.

Scenario 5

You mistakenly log out of Workflow in autologin mode.

Result

The **Login** dialog box appears.

Click the **Autologin** button. The Workflow client logs in with the autologin user credentials.

Scenario 6

You are logged out of Workflow after a predefined period of inactivity.

Result

The **Login** dialog box appears.

Options

- If automatic login is enabled, click **Autologin**. The Workflow client logs in with the autologin user credentials.
- Whether automatic login is enabled or not, enter valid user credentials to log in.
- Click Cancel to close the Proficy client and end the session.

Scenario 7

You logged out of Workflow using the **Log out from Proficy** icon.

Result

The **Login** dialog box appears.

Options

- If automatic login is enabled, click **Autologin**. The Workflow client logs in with the autologin user credentials.
- Whether automatic login is enabled or not, enter valid user credentials to log in.
- Click Cancel to close the Proficy client and end the session.

Scenario 8

The application server is unreachable during Workflow client startup. A message displays stating, "Server is currently unreachable. Check to make sure server is running, or network connection is not broken."

Result

Click **OK**. The Workflow client changes to logged out mode displaying the default login view; however, the **Login** button is not available. When the connection to the server is restored, the **Login** button is enabled.

Click the **Login** button. The Workflow client changes to autologin mode.

Scenario 9

A user is logged in to the Workflow client and the autologin configuration is updated with invalid credentials. When the user logs out, a message displays stating "The user credentials configured for Autologin are invalid. Please ask Administrator to check the configured credentials."

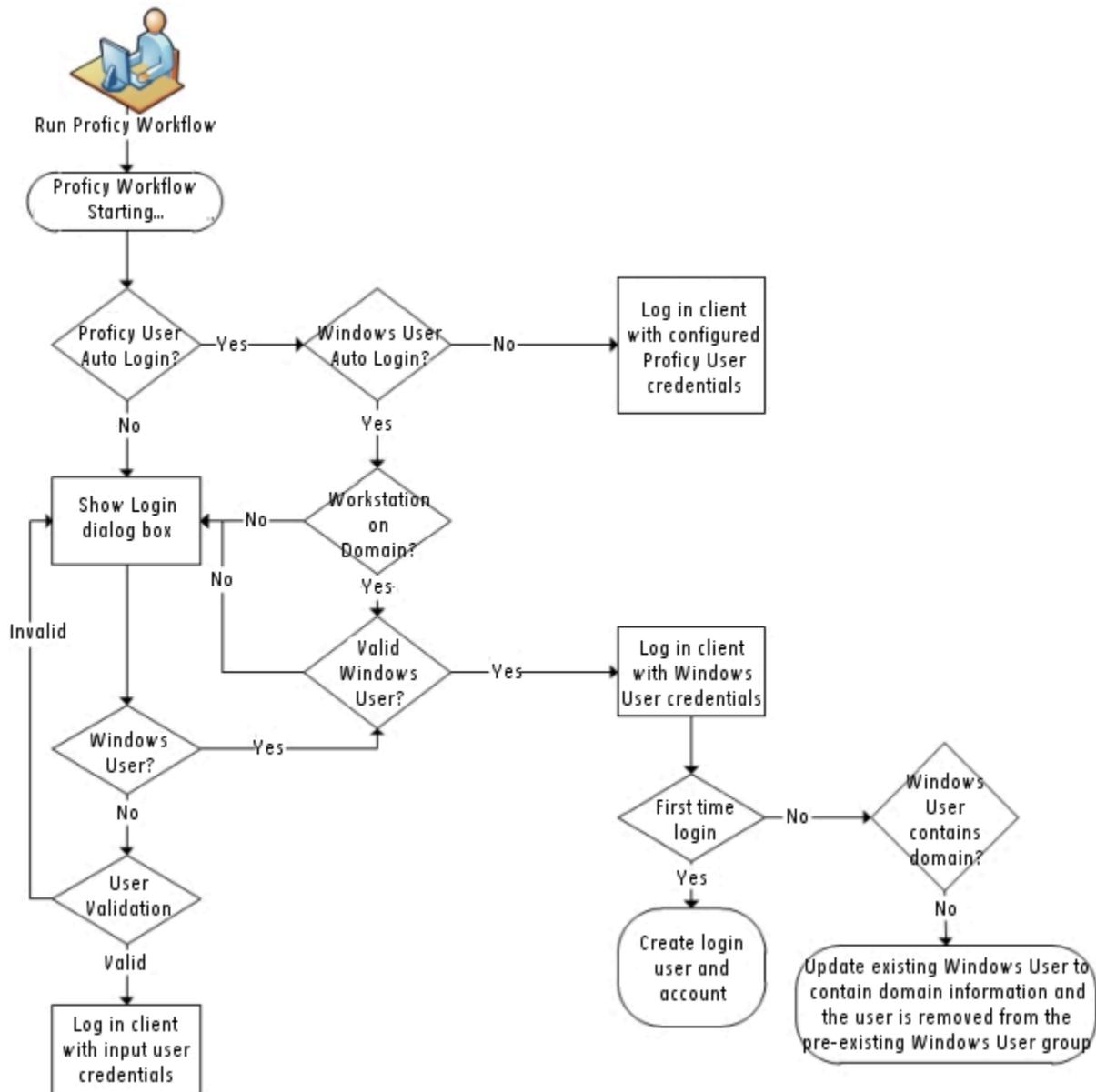
Result

Click **OK**. The Workflow client changes to logged out mode displaying the default login view.

Automatic Login Authentication

When a user logs in to Workflow, his login credentials are authenticated. Workflow user credentials are authenticated by Workflow; Windows user credentials are authenticated by the domain controller.

The following flow diagram illustrates the Workflow automatic login process.



Note:

To log in to Workflow, the system must be in a state of either Complete or Partial health. For more information, see **System Health and Program Use**.

Configure automatic login

Configure automatic login if you want Workflow to automatically log in using the defined Workflow user or the currently logged in Windows user credentials.

Procedure

1. In the navigator, click **Proficy System > Proficy System > Computers**.
2. From the **Computers** list, select the computer that you want to configure.

**Tip:**

If you are configuring settings for all computers in your system or a subset of those computers, you can skip this step. However, you can still configure global default settings if you select an individual computer.

3. In the **Displays** panel, select the editor you want to use.

To configure autologin for...	Select
a Proficy user or Windows users on an individual computer	Computer Editor
Windows users on all computers or a subset of computers in your system	Default Computer Editor

4. Click the **Login/Logout Configuration** tab.
5. In the **Auto Login** section, select the **Enable Auto login** check box.
6. Do one of the following:
 - Select the **Proficy User** option, and then, in the **User Name** and **Password** fields, enter the name and password of the Workflow user who will be automatically logged in when Workflow is started on the selected computer.

**Note:**

This option is available only on the **Computer Editor**.

- Select the **Windows User** option to indicate that automatic login is based on the currently logged in Windows user. That is, when the currently logged in Windows user opens Workflow, he will be automatically logged in.
7. Click **Save**.

**Note:**

Automatic login does not take effect on remote clients until a user has manually logged in once.

What to do next

To log in to Workflow, the system must be in a state of either Complete or Partial health. For more information, see **System Health and Program Use**.

Automatic login after configuration changes

If changes are made to the automatic login configuration, they do not take effect immediately. You must log out and then log back in for the changes to take effect.

Procedure

1. On the Workflow client title bar, click the **Log out from Proficy** icon.
The **Workflow Login** dialog box appears.
2. Click **Cancel**.
The **Workflow Login** dialog box closes.
3. Click the **Log in to Workflow** icon.
4. Click **Autologin**.
The client is logged in with the new automatic login credentials.

What to do next

The next time the client is started, it will automatically log in with those same credentials.

Configure automatic logout

Configure automatic logout if you want Workflow to automatically log out the user after a specified period of inactivity. When you configure automatic logout on a global basis, the specified timeout applies to all computers that are configured to use the global settings.

Procedure

1. In the navigator, click **Proficy System > Proficy System > Computers**.
2. From the **Computers** list, select the computer that you want to configure.



Tip:

If you are configuring settings for all computers in your system or a subset of those computers, you can skip this step. However, you can still configure global default settings if you select an individual computer.

3. In the **Displays** panel, select the editor you want to use.

To configure autologout on...	Select
an individual computer	Computer Editor
all computers or a subset of computers in your system	Default Computer Editor

- Click the **Login/Logout Configuration** tab.
- In the **Auto Logout** section, select the **Enable Auto logout** check box.
- In the **Timeout in** section, enter the number of minutes and/or seconds representing the amount of time that Workflow can remain open without activity before automatically logging the user out of the system.

**Note:**

User activity is defined as moving or clicking the mouse, or pressing keyboard keys.

- Click **Save**.
- Restart the Workflow client for the autologout settings to take effect.

Automatic Login/Logout Configuration

The **Login/Logout Configuration** tab allows you to enable and configure automatic login and logout.

**Note:**

This tab is available in the Computer Editor only if the selected computer is configured to override the global configuration settings.

The following table describes the available options on both the **Computer Editor** and the **Default Computer Editor**. The differences are noted.

Attribute	Value	Description
Enable Auto Login	User-defined	Select this check box to enable automatic login. Clear this check box to disable automatic login.
Proficy User	User-defined	Select this option to indicate that automatic login is based on the defined Workflow user login credentials.

Attribute	Value	Description
		 <p>Note: This option is available only if you are configuring Workflow user accounts on the Computer Editor.</p>
Proficy User: User Name	User-defined	Specifies the name of the user who will be automatically logged in when Workflow is started on this workstation.
Proficy User: Password	User-defined	Specifies the password for this user.
Windows User	User-defined	<p>Select this option to indicate that automatic login is based on the currently logged in Windows user.</p>  <p>Note: This option applies if you are configuring autologin for Windows user accounts on either the Computer Editor or the Default Computer Editor.</p>
Enable Auto Logout	User-defined	<p>Select this check box to automatically log a user out after a specified time if Workflow is inactive on this workstation.</p> <p>You must restart the Workflow client for this setting to take effect.</p>

Attribute	Value	Description
Timeout in	User-defined	Specifies the number of minutes and/or seconds of inactivity allowed before the user is logged out.

Manual Login and Logout

Operators can log into Workflow manually using the **Login** dialog box. When the login dialog box appears, it allows operators to enter their login name and password.



Note:

To log in to Workflow, the system must be in a state of either Complete or Partial health. For more information, see **System Health and Program Use**.

Windows passwords are case sensitive.



Note:

Each time an unsuccessful attempt is made to access the Workflow system, a message is sent to the log file.

Security Log File

Workflow security generates a log of security-related actions taken by Workflow users. The security log file resides in the default log path and is called `ProficySTS.LOG`.

The information in the log file is displayed in chronological order. The security log file is a cumulative file; that is, all new security-based actions are appended to the bottom of the file. To avoid the log file growing too large, it is set to a maximum of 2 MB. If the content grows larger than that, it automatically deletes the oldest information to accommodate the new. You can also manually delete old information from the file.

By reviewing the log file, you can learn about:

- who logged in and out.
- unsuccessful attempts to access Workflow.
- when someone attempted to access an area they had no privileges for.

Workflow User Password Security

Administrator users have the ability to make changes to the security settings in Workflow, including configuring lockout thresholds, re-enabling locked out user accounts, configuring password complexity, and determining whether or not users can change their passwords.

These settings can be configured during the Workflow installation, or at any time after installation. Any changes made to the Administrator credentials or the user authentication settings take effect at the next login session; that is, if a user is currently logged in, changes to these settings will apply after a user logs out and then logs in again.



Important:

When you use the Configure Security tool to make changes to password security, you must re-enter and confirm the Administrator user's password before you can save your changes.

Modify Administrator user account

At any time after installation, you can use the Configure Security tool to change the Administrator user login credentials. If password complexity is enabled and configured, the Administrator password must follow the same rules as defined for user passwords.

About this task



Important:

When you use the Configure Security tool to make changes to password security, you must re-enter and confirm the Administrator user's password before you can save your changes.

Procedure

1. On the server computer, click **Start > All Programs > General Electric > Workflow > Configuration > Configure Security**.
The **Security** page appears.
2. Specify the administrator's authentication credentials.
 - a. In the **Name** field, accept the default name or enter a new name.
 - b. In the **Password** field, enter a password.
 - c. In the **Confirm Password** field, enter the password again.
3. Click **Save**, and then click **Exit**.

Configure GE Single Sign on (SSO)

At any time after installation, you can use the **Configure Security** tool to configure GE Single Sign On (SSO) to be used at your site.

About this task

**Important:**

When you use the Configure Security tool to make changes to password security, you must re-enter and confirm the Administrator user's password before you can save your changes.

Procedure

1. On the server computer, click **Start > All Programs > General Electric > Workflow > Configuration > Configure Security**.
The **Security** page appears.
2. Expand the **Advanced Settings**.
3. Select the **Use SSO (Single Sign On)** check box.

**Note:**

Selecting this check box displays the SSO Authentication option when a user logs on.

4. Select an identity provider option that complies with the level of restrictions and security required for your application.

**Note:**

During a new installation or an upgrade, when you select the Use SSO (Single Sign On) check box, the Production Identity Provider option is selected by default.

**Tip:**

The **Production Identity Provider** option provides greater security for your application.

5. Click **Save**, and then click **Exit**.

Configure Web HMI Access

You can configure Web HMI access during installation or upgrade or use **Configuration Tools** to connect to Web HMI post-installation.

About this task

You can specify a URL to get Web HMI's public key during installation or upgrade, or enter the relevant information post-installation. The public key enables you to login once to GE Digital systems.

If you specify a URL during installation, the installer checks to ensure the URL is valid and accessible.

If you choose to enable Web HMI, a token key is required to authenticate you as a Web HMI user. The key is retrieved from Web HMI and stored in Workflow.

Procedure

- To retrieve a token key during installation or upgrade:
 1. From the **Security** dialog box during installation, do the following:
 - Enter the account name in the **Name** field.
 - Enter a password for the account in the **Password** field, and then re-enter the same password in the **Confirm Password** field.
 - Select the Web HMI Access check box.The **Web HMI Hostname** field appears in the dialog box.
 - Enter the hostname or IP address of the Web HMI server in the **Web HMI Hostname** field.
 2. Click **Next** and proceed with the installation or upgrade.
- To retrieve a token key post-installation or upgrade:
 1. On the server computer, click **Start > All Programs > General Electric > Workflow > Configuration > Configure Security**.
 2. From the **Security** dialog box, do the following:
 - Enter the account name in the **Name** field.
 - Enter a password for the account in the **Password** field, and then re-enter the same password in the **Confirm Password** field.
 - Select the Web HMI Access check box.The **Web HMI Hostname** field appears in the dialog box.
 - Enter the hostname or IP address of the Web HMI server in the **Web HMI Hostname** field.
 3. Click **Save** and then click **Exit**.

Allow users to change account passwords

At any time after installation, you can use the **Configure Security** tool to allow Proficy users to change their password. This capability is available only at login time.

About this task

If you are upgrading from a previous version of Workflow, you must use this tool to configure the password security features. Account lockout is enabled by default during the installation process.



Important:

When you use the **Configure Security** tool to make changes to password security, you must re-enter and confirm the Administrator user's password before you can save your changes.

Procedure

1. On the server computer, click **Start > All Programs > General Electric > Workflow > Configuration > Configure Security**.
The Security page appears.
2. Select the **Allow users to change password** check box if you want to allow users to change their login password at any time.
3. Click **Save**, and then click **Exit**.

Configure Workflow user login limits

At any time after installation, you can use the **Configure Security** tool to configure Workflow user login limits.

About this task

If you are upgrading from a previous version of Workflow, you must use this tool to configure the password security features. Account lockout is enabled by default during the installation process.



Important:

When you use the **Configure Security** tool to make changes to password security, you must re-enter and confirm the Administrator user's password before you can save your changes.

Procedure

1. On the server computer, click **Start > All Programs > General Electric > Workflow > Configuration > Configure Security**.
The **Security** page appears.

2. Select the **Enforce user lockout** check box, and then set the lockout threshold, duration, and timeframe values. This check box is selected by default.
 - a. In the **Lockout threshold** field, enter a number from 1 through 100 to indicate the number of consecutive failed login attempts can be made before the Workflow user account is locked out. The default lockout threshold is 5.
 - b. In the **Lockout duration** field, enter a number from 0 through 10,000 to indicate that the locked out account will remain locked out for the defined number of minutes. A value of 0 means that the account is locked out indefinitely and must be reactivated by an Administrator user. The default lockout duration is 30 minutes.
 - c. In the **Lockout timeframe** field, enter a value from 1 through 10,000 to indicate that the failed login count will be reset after the defined number of minutes after the last failed login attempt. A successful login after this duration has passed resets the window to this value. The default window size is 30 minutes.
3. Click **Save**, and then click **Exit**.

Configure Workflow user password complexity rules

At any time after installation, you can use the **Configure Security** tool to configure Workflow user password complexity rules. Complexity rules do not affect existing passwords. However, if a user changes his password, the complexity rules will apply.

About this task

If you are upgrading from a previous version of Workflow, you must use this tool to configure the password security features. Account lockout is enabled by default during the installation process.



Important:

When you use the **Configure Security** tool to make changes to password security, you must re-enter and confirm the Administrator user's password before you can save your changes.

Procedure

1. On the server computer, click **Start > All Programs > General Electric > Workflow > Configuration > Configure Security**.
The **Security** page appears
2. Select the **Enforce user password complexity rules** check box, then select Simple, Normal, or Advanced to determine the level of complexity for your passwords.

- Select **Simple** to specify that a minimum of six characters of any type must be used in user passwords.
 - Select **Normal** to specify that a minimum of six characters, composed of at least two digits (0-9) and three letters (a-z and/or A-Z), must be used in user passwords.
 - Select **Advanced** to configure complex password rules. Each rule is enforced separately. Valid values range from 0 (disabled) through 100.
3. Click **Save**, and then click **Exit**.

Password Security Configuration

When you install Workflow, you are required to set up an Administrator user and you have to option to configure user authentication settings. At any time after installation, you can use the **Configure Security** tool to change the Administrator user login credentials, GE Single Sign On (SSO), enable users to change their password, configure login limits, and configure password complexity.

Administrator User Configuration

The following table describes the information available for configuring the Administrator user credentials.

Property	Value	Description
Name	User-defined	Specifies the login name for the Administrator user. The default is Admin.
Password	User-defined	Specifies a unique password for the Administrator user.
Confirm Password	User-defined	Specifies the same unique password that you entered in the Password field.

GE Single Sign On (SSO)

When GE Single Sign On is used, this option becomes available on the login dialog box.

The following table describes the information available for enabling the use of GE Single Sign On logins.

Property	Value	Description
Use SSO	N/A	Indicates whether Workflow uses GE Single Sign On for login purposes. For an upgrade installa-

Property	Value	Description
		tion, you must use the Configure Security tool to enable the GE Single Sign On option on the login dialog box.
Production Identity Provider	User-defined	Indicates that you require greater restrictions and security for your environment. This option is recommended for both production and non-production environments. During a new installation or an upgrade, when you select the Use SSO (Single Sign On) check box, the Production Identity Provider option is selected by default.
Non-Production Identity Provider	User-defined	Indicates that you do not require greater restrictions and security for your environment. This option can be used for non-production environments, such as test environments.

Workflow User Password Change

The following table describes the information available for enabling the ability to change passwords.

Property	Value	Description
Allow Password Change	N/A	Indicates whether Workflow users can change their passwords when they log in to Workflow. For an upgrade installation, you must use the Configure Security tool to enable the change password link on the login dialog box.

Account Lockout Settings

If a user exceeds the defined login limit, his account is locked out and he must either wait for the lockout duration to pass or ask an Administrator user to reactivate his account.

The following table describes the information available to configure Workflow user login limits.

Attribute	Value	Description
Enforce User Lockout	User-defined	Select this check box to indicate that account lockout rules are being used. This check box is selected by default during a full installation and during an upgrade installation. For an upgrade installation, you must use the Configure Security tool to disable account lockout settings.
Lockout threshold	User-defined	Specifies the number of consecutive failed login attempts that can be made before a Workflow user account is locked out of Workflow. Valid values are 1 through 100. The default lockout threshold is 5.
Lockout duration	User-defined	Specifies the number of minutes that a Workflow user account is locked out of Workflow when it fails to successfully log in the defined number of times. Valid values are 0 through 10,000. A value of 0 indicates that an account is locked out indefinitely and must be reactivated by an Administrator user. The default lockout duration is 30 minutes.
Lockout timeframe	User-defined	Specifies the number of minutes after the last failed login attempt

Attribute	Value	Description
		occurs before the failed login count is reset and the user can attempt to log in again. A successful login after this duration has passed resets the window to this value. Valid values are 1 through 10,000. The default window size is 30 minutes.

Example

If you define a threshold of 5, duration of 60 minutes, and observation window size of 30 minutes then, if a user attempts to log in and fails five time within 30 minutes, his account will be locked out for 60 minutes. All login attempts for that account will be rejected until the 60 minutes is up or an Administrator user unlocks the account.

Password Complexity Rules

Password rules are enforced on new passwords and when a user's password is changed, and each rule is enforced separately. The values for each rule can range from 0 (disabled) to 100.

The following table describes the information available for configuring password complexity.

Rule	Value	Description
Enforce Password Complexity Rules	N/A	Select this check box to indicate that password complexity rules are being used. This check box is selected by default during a full installation. For an upgrade installation, you must use the Configure Security tool to enable password complexity rules.
Simple	N/A	Select this option to indicate that user passwords must contain at least six characters of any type.
Normal	N/A	Select this option to indicate that user passwords must contain at

Rule	Value	Description
		least six characters and those characters must be composed of at least two digits (0-9) and three letters (a-z and/or A-Z).
Advanced	N/A	Select this option to configure complex password rules. Each rule is enforced separately. Valid values range from 0 (disabled) through 100.
Advanced: Minimum length of password	User-defined	The minimum number of characters of any type that a user's password can contain.
Advanced: Minimum number of alphanumeric characters	User-defined	<p>The minimum number of alphanumeric characters a user's password can contain. Valid values are:</p> <ul style="list-style-type: none"> • a-z • A-Z • 0-9
Advanced: Minimum number of numeric characters	User-defined	The minimum number of numeric characters a user's password can contain. Valid values are 0-9.
Advanced: Minimum number of alphabet characters	User-defined	<p>The minimum number of alphabet characters a user's password can contain. Valid values are:</p> <ul style="list-style-type: none"> • a-z • A-Z
Advanced: Minimum number of lowercase characters	User-defined	The minimum number of lowercase alphabetical characters a user's password can contain. Valid values are a-z.

Rule	Value	Description
Advanced: Minimum number of uppercase characters	User-defines	The minimum number of uppercase alphabetical characters a user's password can contain. Valid values are A-Z.
Advanced: Minimum number of non-alphanumeric characters	User-defined	The minimum number of non-alphanumeric characters a user's password can contain. Non-alphanumeric characters include special characters, such as #, %, @, and _.

Chapter 7. Resource Information and Configuration

Resource Information and Configuration

Resource Information and Configuration

This section describes the models and other resources that must be configured before you can use Workflow.

Materials are the substances or components that are used to make something else. A material can be a finished product in its own right, or an unprocessed raw material.

The Inventory View provides read-only information on material definition quantities in lots and descendant sublots associated with material definitions and equipment resources.

The equipment model presents a structured hierarchy of resources to develop a modular representation of a complete production or manufacturing organization in a physical and functional context. The model is flexible, addressing batch, continuous, and discrete production types, or representing inventory storage.

The production model presents a set of resources used to describe a product as a collection of components and the processes applied to those components.

Events are a fundamental part of production. The events model is used to map out the critical events that occur to control production. These events can be created within this application or from external systems, and can be used to control workflow execution.

The Event View displays multiple active condition and time events in the Proficy System and the resources that currently are attached to the events.

Resource actuals show the as-performed actual values for a resource during a production cycle. A resource actual is associated with a resource specification, making it easy to compare actual and specified values for properties.

Property groups are containers for properties. They allow you to read from or write to the properties of a resource.

Aspects are a mechanism for dynamically extending existing resources by adding new attributes and business logic without modifying the original objects or recompiling them. They can be used to extend the attributes of an existing object, or they can link two or more disparate entities.

Material Model

Material Model

Materials are the substances or components that are used to make something else. A material can be a finished product in its own right, or an unprocessed raw material.

The material model is made up of material definitions and material classes that are used in the manufacturing or production of a product.

Material specifications are not created as material resources, but are material definition or class resources that are associated with work process segments.

**Note:**

You can configure up to 5000 properties for each material definition and 1000 properties for each material class, and you can add up to 50 material classes to each material definition for a maximum of 55000 properties per material definition.

Add properties or parameters to a resource

Procedure

1. From the navigator, select the model that contains the resource you want to work with.
In the next pane, a list of resources appears.
2. Select the resource you want to work with.
3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click the **Properties** or **Parameters** tab, and then click **Add Property** or **Add Parameter**.
5. In the **Name** and **Description** fields, enter a name and description for the property/parameter.
6. From the **Data Type** list, select a data type.
7. In the **Value** field, enter a value for the selected data type.
8. **Optional:** If applicable, in the **Unit of Measure** field, enter a unit of measure for the selected data type.
9. Click **Save**.

Make a material property a template for a material lot property

Before you begin

Only material definition properties can be used for material lot property templates.

This option is cleared by default.

Procedure

1. From the navigator, select the model that contains the resource you want to work with.

In the next pane, a list of resources appears.

2. Select the resource you want to work with.
3. In the **Displays** panel, select an editor.

The editor appears in the workspace, displaying the resource information.

4. Click the **Properties** tab, and then select the property or parameter you want to make a template.
5. Select the **Template for Material Lot Property** check box.
6. Click **Save**.

Add a material class to a material definition

Before you begin



Note:

You must have added at least one material class resource. For more information, see **Add a resource to the material model**.

Procedure

1. In the navigator, click **Material**.
2. In the **Folders** list, select the resource type you want to work with.



Tip:

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.

The editor appears in the workspace, displaying the resource information.

4. Click the **Classes** tab.
5. Click **Add Class**.

The **Universal Browser** appears.

6. Select the material class you want to add, and then click **OK**.

**Tip:**

To remove a material class, from the classes list, select a class, and then click **Delete Selection**.

7. Click **Save**.

Use custom property types in the material model

Procedure

1. From the navigator, select **Materials > Definitions**.
2. From this list, select the resource you want to work with.
3. In the Displays panel, click **Production Management > S95 Model Editor**.
4. Click the **Properties** tab, and then click **Add Property**.
5. In the **Name** field, enter a name for this property.
6. From the **Type** field, select a custom property type.
7. In the **Value** field, select a value for the custom property type.
8. **Optional:** If applicable, in the **Unit of Measure** field, enter a unit of measure for the custom property type.
9. Click **Save**.

Material Classes

A material class is a grouping of common properties that can be applied to material resources.

Material classes can be directly associated with work process segments only. When you associate a work process segment with a work definition segment, the associated material class, and its properties, are made available to the work definition segment, and then to any work definitions that the work definition segment is associated with. A material class is distinguished from other classes through its properties, name, and description.

You can create a material class for paint. The class defines the paint by its individual components that need to be mixed together. To define paint as a material class, include all the corresponding elements for the paint that are needed, other than the color. By not including the color, the class can now be used with different colors of paint. The color itself is then defined by the material definition, process segment, or product segment.

Related reference

[Material Definitions \(on page 413\)](#)

Material Classes Details

The **Details** tab offers general information about the selected material class.

You can change the name and description of this resource.

Attribute	Value	Description
Name	Required; user-defined up to 50 characters	Specifies a display name for a resource.
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

Material Classes Properties

The **Properties** tab allows you to add, modify, and delete properties that are specific to the selected material class.

The following table describes the information available on the Properties tab, where you can configure the values for the property attributes.

Attribute	Value	Description
List	User-defined	Displays a list of properties for the selected resource that can be arranged by name, type, and so on, depending on the property. These properties can be inherited from another resource or created for the current resource.
Add Property	N/A	Click to add a new property.
Delete Property	N/A	Click to remove the selected property.
Name	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 50 characters	Specifies a display name for the selected property.

Attribute	Value	Description
Description	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 255 characters	Provides more information about a property that a user may find helpful.
Type	System-defined	<p>Specifies the data type of the property.</p> <div data-bbox="1024 537 1422 1371" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note:</p> <ul style="list-style-type: none"> This property cannot be changed if it is inherited from another resource. When you select Browse for class, the Universal Browser appears where you can select an equipment class. This field is not available if the Linked check box is selected. </div>
Value	User-defined alphanumeric string (A-Z, a-z, 0 - 9) up to 8000 characters	Specifies the value for the selected data type. The options available depend on the selected data type.
UoM (Unit of Measure)	User-defined alphanumeric string (A-Z, a-z, 0 - 9) up to 20 characters	Specifies the unit of measure for the data type, such as gallons, inches, tons.

Material Definitions

A material definition describes supplies or materials with unique characteristics. Definitions offer a specific defined material to be used in a production process in the creation of a final product.

A material definition can have multiple material classes associated with it, and inherits all of the properties defined for those material classes. You can also add properties directly to the material definition. A material definition can also stand alone with no associated material classes.

A material definition does not have to be a physical object, but can be characteristics of an object.

An example of a material definition is all of the components used to create the seat of a bicycle.

Related reference

[Material Classes \(on page 410\)](#)

[Material Lots and Sublots \(on page 416\)](#)

[Material Inventory \(on page 417\)](#)

Material Definition Details

The **Details** tab offers general information about the selected material definition.

You can change the name and description of this resource.

Attribute	Value	Description
Name	Required; User-defined up to 50 characters	Specifies a display name for a resource.
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

Material Definition Properties

The **Properties** tab allows you to add, modify, and delete properties that are specific to the selected material definition, and to view the properties for any associated material classes.

The following table describes a property specific to material definitions:

Property	Value	Description
Template for Material Lot Property	N/A	Select this check box to make this property available in a material lot.

The following table describes the information available on the **Properties** tab, where you can configure the values for the property attributes.

Attribute	Value	Description
List	User-defined	Displays a list of properties for the selected resource that can be arranged by name, type, and so on, depending on the property. These properties can be inherited from another resource or created for the current resource.
Add Property	N/A	Click to add a new property.
Delete Property	N/A	Click to remove the selected property.
Name	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 50 characters	Specifies a display name for the selected property.
Description	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 255 characters	Provides more information about a property that a user may find helpful.
Type	System-defined	Specifies the data types of the property. <div data-bbox="1024 1556 1421 1837" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note:</p> <ul style="list-style-type: none"> This property cannot be changed if it is inherited from another resource. </div>

Attribute	Value	Description
		 <ul style="list-style-type: none"> • When you select Browse for class, the Universal Browser appears where you can select an equipment class. • This field is not available if the Linked check box is selected.
Value	User-defined alphanumeric string (A-Z, a-z, 0 - 9) up to 8000 characters	Specifies the value for the selected data type. The options available depend on the selected data type.
UoM (Unit of Measure)	User-defined alphanumeric string (A-Z, a-z, 0 - 9) up to 20 characters	Specifies the unit of measure for the data type, such as gallons, inches, tons.

Material Definitions Classes

The **Classes** tab allows you to associate one or more material classes with the selected material definition.

Classes are like property templates that are selectively applied to specific resources. Multiple classes can be applied to a given resource. For example, if a set of properties applies to multiple equipment locations, you can create a class that contains all of these properties and then add that class to each of the applicable equipment locations.

The following table describes classes attributes:

Attribute	Value	Description
Class List	User-defined	Displays a list of classes that have been added to the selected resource.

Attribute	Value	Description
Add Class	N/A	Click to open the Universal Browser to select the class you want to add to the selected resource.
Delete Selection	N/A	Click to remove the selected class from the resource.

Material Lots and Sublots

A lot or a subplot is a quantity of a material definition and is associated with an equipment resource. Lots can be viewed in the **Inventory View** of an associated material definition or associated equipment instance. The **Inventory View** is lot-based and sublots are visible only under their associated lot.

Lots and sublots can be directly associated with one material definition and one equipment resource only. A lot or subplot is distinguished by its S95Id. The description, material definition, location (equipment association), status, quantity, and unit of measure may be identical for lots and sublots.

Sublots are placed below lots and other sublots to create a hierarchy. Sublots can use distinct locations, statuses, quantities, and units of measure, but must share the same material definition as the lot. The lot or parent subplot quantity is not a sum of the child subplot quantities.

You can create a lot for the material definition: paint. The lot provides an inventory of all paint quantities subdivided in a hierarchy you design. Each lot and subplot has the properties defined as lot properties in the material definition.

Properties

Lot properties are derived from the associated material definition. In order for a material definition property to be associated with a lot, the **Template for Material Lot Property** check box must be selected on the material definition's **Properties** tab.

Related reference

[Material Inventory \(on page 417\)](#)

[Material Definitions \(on page 413\)](#)

[Material Definition Properties \(on page 413\)](#)

Related information

[Equipment Model \(on page 421\)](#)

[Make a material property a template for a material lot property \(on page 408\)](#)

Material Inventory

Material Inventory

The **Inventory View** provides read-only information on material definition quantities in lots and descendant sublots associated with material definitions and equipment resources.

The following table describes the properties available in the Inventory View.



Note:

The **Inventory View** displays lots associated with a selected material definition or equipment model. Sublots appear under their associated lot only, even those with different equipment model associations.

Property	Description	
S95 ID		Specifies the name of a lot or subplot.
Description		Specifies the description of the lot or subplot.
Material Definition		Specifies the material definition that is associated with the lot. <div data-bbox="1024 1188 1419 1455" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note: All sublots must share the same material definition as their parent lot or subplot.</p> </div>
Location		Specifies the equipment resource associated with the lot or subplot.
Status		Specifies the state of the lot or subplot and associated resources.
Quantity		Specifies the quantity of the material in the lot or subplot.

Property	Description	
		 Note: Lots and parent subplot quantities are not the sum of all child subplot quantities. All quantities of lots and sublots are independently set.
UoM (Unit of Measure)		Specifies the unit of measure for the quantity of material.  Note: Sublots can have different units of measure than their parent lot or subplot.

Properties For

You can view the associated material definition properties of a lot or subplot in the **Properties For** panel. The S95Id of the selected lot or subplot appears in the panel heading.

Click the arrow button to show or hide the panel.

The following table describes the property attributes that are visible:

Attribute Name	Description	
Property Name		Specifies the display name for a resource property.
Value		Specifies the value of the property.
Units		Specifies the unit of measure for the property, such as gallons, inches, or tons.

Related reference[Material Lots and Sublots \(on page 416\)](#)[Material Definitions \(on page 413\)](#)**Related information**[Equipment Model \(on page 421\)](#)[Make a material property a template for a material lot property \(on page 408\)](#)

Filter material lots

About this task

The **Inventory View** uses filtering to retrieve and display large amounts of data in the Workflow client workspace.

Procedure

1. In the **Inventory View** editor, click **Show Filters**.
The filter editor appears.
2. To filter material lots by date, click the drop-down arrow on **Start Date** and/or **End Date** and set the required date(s) and time(s).
3. To filter material lots by quantity, enter the minimum and maximum number of material lots for display.
4. To filter material lots by name, enter the S95 ID of the material lot to display.
5. To filter lots by location, click the ellipsis button, and then select the location.
6. To filter lots by status, enter the status in the **Status** field.
7. To select all closed material lots, select the **Closed** check box.
8. Click **Refresh**. The view changes to the **Inventory View** editor where the filtered material lots are listed.

Material Inventory Filters

The **Inventory View** uses filtering to retrieve and display large amounts of data more efficiently.

Clicking the **Show Filters** button opens the filter editor, which contains filtering options for material lots.

The following table describes the available filtering options:

Attribute	Value	Description
Start Date	DateTime	Specifies the earliest date and time to filter by. Only lots that oc-

Attribute	Value	Description
		curred on or after this date are returned.
End Date	DateTime	Specifies the latest date and time to filter by. Only lots that occurred on or before this date are returned.
Minimum	Int32	Specifies the minimum number of lots to display.
Maximum	Int32	Specifies the maximum number of lots to display.
S95 ID	String	<p>Specifies the name of the material lot to filter by.</p> <div data-bbox="1024 898 1419 1260" style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note: The name you enter must be an exact match. S95 IDs are case sensitive and wild cards are not permitted in a filter query.</p> </div>
Location	DirectoryResource	Specifies the location of the material lot to filter by.
Status	String	Specifies the status to filter by.
Closed	Boolean	Select this check box to return material lots that are closed.

When you use filter parameters, the relevant material lots are displayed in the **Inventory View**. If the material lot set returns a greater number than the maximum quantity permits, a link appears in the results display. Clicking this link advances the display to show the next set of material lots within the allowable range.

If a material lot is expanded and it contains more results than the maximum quantity permits, a link appears in the results display. Clicking this link retrieves the next set of material sublots within the allowable range.

If you click the **Refresh** button with no filter selected, the original set of material lots is displayed.

Equipment Model

Equipment Model

The equipment model presents a structured hierarchy of resources to develop a modular representation of a complete production or manufacturing organization in a physical and functional context. The model is flexible, addressing batch, continuous, and discrete production types, or representing inventory storage.

The permissible hierarchies depend on the process being modeled, which includes batch, continuous production, discrete production, storage, or movement. However, within these hierarchies, an equipment resource may also own other resources of the same type. For example, production lines can own other lines; units can own sub-units, and so on. You can use these basic types to model the equipment within your enterprise, adding properties for each resource, and creating equipment classes. The S95 Editor provides a user interface to model equipment within your project.

The hierarchies within the Equipment model have two types of resources: intermediate and terminating. Intermediate resources are resources that exist between levels of the hierarchy. For example, a site that exists between an enterprise and an area. Terminating resources are resources that are the last resource in a hierarchy.

In addition to the hierarchies, alternative production paths are possible. For example, your production hierarchy may have three lines that can perform the same function: two active (lines 1 and 2), and one backup (line 3). If production stops on line 2, you can choose to use an alternative path to line 3 while line 2 is repaired.

**Note:**

You can configure up to 5000 properties for each equipment instance, inclusive of the properties of any equipment classes that may be associated with the instance. For example, if an equipment instance has five classes associated with it, and each of those classes has 100 properties, then the instance could have 4500 properties configured at the instance level.

Related reference

[Aspects \(on page 559\)](#)

[Material Lots and Sublots \(on page 416\)](#)

[Material Inventory \(on page 417\)](#)

[Equipment Actuals \(on page 453\)](#)

Generate an Equipment Model for Web HMI

You can generate an Equipment Model CSV file from Workflow data that can then be uploaded to Web HMI.

About this task

You can select equipment items to be excluded from the CSV file prior to generation. Once generated, the saved CSV file can be edited and uploaded to the Web HMI application.

Procedure

1. In the navigator, select **Proficy System > Proficy System**.
2. From the **Displays** panel, select **HW/SW > Generate Operations Hub Model**.
The **Generate Operations Hub Model** page appears.
3. Set up a path for a new CSV file as follows:
 - a. Under **Configuration**, click the ellipsis (...) for the **Output File Path** field.
 - b. Navigate to a folder on your local drive.
 - c. Enter a name for the CSV file after the path in **File Name**.
 - d. Select **CSV File (.csv) (*.csv)** in the **Save as type** field, and then click **Save**.
The selected path appears in the **Output File Path** field.
4. **Optional:** If you want to exclude any portion of the **Equipment Model** in your generated file, do the following:
 - a. Click **Select**.
The **Universal Browser** appears.
 - b. Navigate through the columns and select one or more items from the hierarchy to exclude.

**Tip:**

Use `Shift+click` to select continuous items or `Ctrl+click` to select individual items.

c. Click **OK**.

The item(s) appears next to **Select**.

**Note:**

If you want to remove an equipment item from the exclusion list, click the **X** next to the item.

5. Click **Generate**.

The model file successfully generates and opens on the desktop.

Generate an Equipment Model for Operations Hub Using GenerateWebHMIModel.exe

You can generate multiple equipment models for Operations Hub that use different export configurations files coming from different systems.

About this task

You can generate an Equipment Model CSV file from Workflow data that can then be uploaded to Operations Hub. You can select equipment items to be excluded from the CSV file prior to generation.

Once generated, the saved CSV file can be edited and uploaded to the Operations Hub application.

**Tip:**

You can create and run scripts that will generate multiple files for multiple projects.

Procedure

1. On the Workflow server, open a command prompt (cmd).
2. Navigate to `C:\Program Files (x86)\Proficy\Proficy Workflow\Program`.
3. Enter `GenerateWebHMIModel.exe` to generate a CSV file using default values.

Launching **GenerateWebHMIModel.exe** followed by the `-help` parameter lists all supported

```

Microsoft Windows [Version 10.0.16299.371]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\lgeftest>cd C:\Program Files (x86)\Proficy\Proficy Workflow\Program
C:\Program Files (x86)\Proficy\Proficy Workflow\Program>GenerateWebHMIModel.exe -help
This tool is used to generate a Web HMI object import file based on the
Workflow equipment model.
Arguments may be prefixed by '/' or '-'

Usage: GenerateWebHMIModel.exe [-DatabaseName <String>] [-DatabasePassword
<String>] [-DatabaseServer <String>] [-DatabaseUserName <String>]
[-ExcludeEquipmentIds <Guid>...] [-ExcludeEquipmentNames <String>...]
[-Help] [-LogFile <String>] [-OutputFile <String>]

-DatabaseName <String>
    Name of the Workflow database to export from. Default is SOADB. Alias:
    -db.

-DatabasePassword <String>
    SQL Server user password if not using integrated security. Alias: -pwd.

-DatabaseServer <String>
    Database Server hosting the Workflow database. Default is localhost.
    Alias: -server.

-DatabaseUserName <String>
    SQL Server user name if not using integrated security. Alias: -user.

-ExcludeEquipmentIds <Guid>
    List of comma separated equipment ids to exclude from the export. Also
    excludes all children of the specified equipment. Alias: -excludeid.

-ExcludeEquipmentNames <String>
    List of comma separated equipment names to exclude from the export.
    Also excludes all children of the specified equipment. Name must be
    unique in the Workflow model. Alias: -excludename.

-Help [<Boolean>]
    Displays this help message. Alias: -?.

-LogFile <String>
    Full path to the log file. Alias: -log.

-OutputFile <String>
    Full path to the output CSV file. Alias: -out.
    
```

commands.

4. If you intend to generate multiple files, you must navigate to the CSV file and change its name. This prevents the file from being overwritten.

The default name for the CSV output is **WebHMIModel.csv**.

Equipment Duplication

While Workflow allows for infinite levels of equipment in its hierarchy, Plant Applications has only three levels in the equipment model.

Workflow	Plant Applications
Enterprise	No equivalent
Area	No equivalent
Process Cell	Department
Production Line	Production Line
Production Unit	Production Line
Storage Zone	Production Line
Any equipment below	Production Unit

When you try to duplicate a production unit or other equipment instance lower in the hierarchy from within the navigator, it can result in an aspect error because the children are moved to the same level as the original instance and this may result in duplicate names.

To avoid this when duplicating an equipment instance, you can specify a prefix to be applied to the names of all duplicated child equipment to ensure unique names in the model. For example, if you specify a prefix of 'xyz,' the child equipment will be named xyzChild1, xyzChild2, and so on. This extends all the way down the equipment hierarchy.

To not use a prefix, leave the prefix field empty.

Duplicate an equipment instance

You can duplicate any equipment resource.

About this task

When you duplicate an equipment instance, by default, its parent is the same as the original, except for an enterprise instance, which is at the top of the hierarchy and has no parent. You cannot change the parent of an enterprise instance. All the child instances of the original instance are also duplicated.

Procedure

1. In the navigator, click **Equipment > Instances**.
2. Select the equipment instance that you want to duplicate, and then click **Duplicate**.
3. Enter a name and description for the new instance.
4. **Optional:** If required, specify a prefix to be applied to the names of all of the instance's duplicated child equipment to ensure unique names in the model.
5. **Optional:** To move the duplicated instance to a different parent, click **Change**, and then from the **Universal Browser**, select a new parent.



Note:

A parent can be any instance type that the duplicate can be a child of. For example, if you duplicate a production line, the new parent can be an area or another production line.

6. Click **OK**, and then click **Save**.

Change the parent equipment for an equipment resource

Before you begin

This procedure does not apply to Enterprises and Classes.

Procedure

1. In the navigator, click **Equipment > Instances**.
2. Continue selecting from the lists that appear until you find the resource you want to work with.

**Note:**

All areas that are compatible with the current resource appear. Green icons indicate a compatible view.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. On the **Details** tab, click **Change**.
The **Universal Browser** appears.
5. Select the resource you want to use as the parent equipment. If more lists appear, continue to select the appropriate resource until you find the resource you want to work with.
6. Click **OK**, and then click **Save**.

Add an equipment class to an equipment definition

Procedure

1. In the navigator, click **Equipment > Instances**.
2. Continue selecting from the lists that appear until you find the resource you want to work with.

**Note:**

All areas that are compatible with the current resource appear. Green icons indicate a compatible view.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click the **Classes** tab.
5. Click **Add Class**.
The Universal Browser appears.
6. Select the equipment class you want to add, and then click **OK**.

**Tip:**

To remove an equipment class, select the class, and then click **Delete Selection**.

7. Click **Save**.

Add properties to a resource

Procedure

1. In the navigator, click **Equipment > Instances**.
2. Continue selecting from the lists that appear until you find the resource you want to work with.
3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click the **Properties** tab, and then click **Add Property**.
5. In the **Name** field, enter a name for this property.
6. In the **Type** field, select a data type or browse for a class.



Note:

If you selected an equipment class as the data type, you must expand the property to view the class properties.

7. In the **Value** field, enter a value for the selected data type.
8. If applicable, in the **Unit of Measure** field, enter a unit of measure for the selected data type.
9. To bind the property to an external data source, do the following:



Note:

You cannot bind equipment class properties to an external data source.

- a. Select the **Linked** check box.
 - b. Click in the **Value** field, and then click the ellipsis button.
The Configure Data Source dialog box appears.
 - c. Configure the data source, as required.
 - d. If applicable, click the **Show Details** button, and then click **Test external data link** to test whether the data source is valid.
10. In the **Description** field, enter a description for this property.
 11. Click **Save**.

Use custom property types in the equipment model

Procedure

1. From the navigator, select the model that contains the resource you want to work with.
In the next pane, a list of resources appears.

2. From this list, select the resource you want to work with.
3. In the Displays panel, click **Production Management > S95 Model Editor**.
4. Click the **Properties** tab, and then click **Add Property**.
5. In the **Name** field, enter a name for this property.
6. From the **Type** field, select a custom property type.
7. In the **Value** field, select a value for the custom property type.
8. **Optional:** If applicable, in the **Unit of Measure** field, enter a unit of measure for the custom property type.
9. Click **Save**.

Copy and paste property fields

Procedure

1. In the navigator, click **Equipment > Instances**.
2. Continue selecting from the lists that appear until you find the resource you want to work with.
3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click the **Properties** tab.
5. Select the property attribute (cell) you want to copy, and then press `Ctrl+C`.
6. Add a new property, or select an existing property, select the same attribute in the property you want to paste the copied information to, and then press `Ctrl+V`.



Note:

- If the copied content is not valid, an error dialog box appears. Click **OK** to cancel the paste action.
- You cannot copy or paste information from an inherited property.

The copied information is pasted to the selected cell.

7. Click **Save**.

Modify the values for a property

Procedure

1. In the navigator, click **Equipment > Instances**.
2. Continue selecting from the lists that appear until you find the resource you want to work with.
3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click the **Properties** tab.

5. Change the property's attributes, as required.

**Note:**

- In the Type field, selecting Browse for class opens the Universal Browser, where you can select a class.
- Changing property values affects your production process.

6. Click **Save**.

Delete properties from a resource

Before you begin

**Note:**

When a property is deleted, it cannot be recovered.

Procedure

1. In the navigator, click **Equipment > Instances**.
2. Continue selecting from the lists that appear until you find the resource you want to work with.
3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click the **Properties** tab.
5. Select the property you want to delete, and then click **Delete Property**.
6. Click **Save**.

Configure the columns for the properties tab

Procedure

1. In the navigator, click **Equipment > Instances**.
2. Continue selecting from the lists that appear until you find the resource you want to work with.
3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click the **Properties** tab.
5. Click **Configure Columns**.
The Configure Columns dialog box opens.
6. Select the check box for the column you want to make visible in the Properties tab or clear the check box for a column to hide it.
7. Click **OK**.
8. Click **Save**.

Order the properties in the properties tab

Procedure

1. In the navigator, click **Equipment > Instances**.
2. Continue selecting from the lists that appear until you find the resource you want to work with.
3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click the **Properties** tab.
5. In the name field of a property with sub-properties, click the expand/collapse arrow to show or hide the sub-properties.
6. Click in the title field for a column.



Note:

When the grid is sorted by anything other than name, the property names are fully qualified; each property and sub-property has its own row with no hierarchy visible. When displayed as fully qualified, no property can be expanded to show sub-properties or closed to hide sub-properties.

The properties are organized alphabetically in that column.

7. Click the same column title again to reverse the order of the properties.
The ordering of properties is replicated for all equipment and equipment class **Properties** tabs.

Enterprises



An enterprise resource represents a complete production organization and is the highest level resource in the equipment hierarchy.

Enterprises are independent because they are not subordinate to any other resource and they are the originators of production planning for a coordinated group of sites or areas.



Note:

Within the enterprise resource, you must have a site defined where child resources are configured to further define your process.

In the equipment model, the enterprise is the topmost resource, and must contain at least one site resource.

The following information describes the hierarchy of this resource.

Required:	one or more
Contained by:	(none)
Contains:	one or more Sites

The following sections describe the information and properties available on each of the tabs on the S95 Editor for equipment resources.

Details Tab

You can change the name and description of this resource.

Attribute	Value	Description
Name	Required; user-defined up to 50 characters	Specifies a display name for a resource.
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

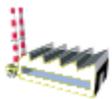
Properties Tab

For more information see **Equipment Properties**.

Classes Tab

For more information, see **Equipment Classes**.

Sites



A site resource represents a geographic subdivision of a production organization, with self-contained production capabilities.

A site can be any high-level grouping of equipment representing a physical or logical organization established by the enterprise. A site can contain another site; that is, it is an intermediate resource that can contain other equipment resources.

The following information describes the hierarchy of this resource:

Required:	one or more
Contained by:	Enterprise
Contains:	one or more Areas

Details Tab

You can change the name and description of this resource.

Attribute	Value	Description
Name	Required; user-defined up to 50 characters	Specifies a display name for a resource.
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

Properties Tab

For more information, see **Equipment Properties**.

Classes Tab

For more information, see **Equipment Classes**.

Areas



An area represents a group of physical production equipment.

Resources contained by an area define where the physical production occurs. Areas correspond to specific models of continuous, discrete, or batch manufacturing. To further define production, multiple areas can be used to define the required process.

An area is an intermediate resource that can contain other equipment resources.

The following information describes the hierarchy of this resource:

Required:	one or more
Contained by:	Site

Contain:**Production lines; Process cells; Production units;
Storage zones**

Details Tab

You can change the name and description of this resource.

Attribute	Value	Description
Name	Required; user-defined up to 50 characters	Specifies a display name for a resource.
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

Properties Tab

For more information see **Equipment Properties**.

Classes Tab

For more information, see **Equipment Classes**.

Production Units



A production unit represents a group of physical equipment items applied to a continuous manufacturing process.

Production units are self-sufficient and are not dependent on external equipment items or processes to carry out the portion of a continuous production process they are defined for. It is a resource that can be the last resource in a hierarchy; that is, a terminating resource. Although a production unit can encompass multiple physical equipment items, those items (for example, equipment that converts, separates, or reacts) are not represented in the equipment model.

The following information describes the hierarchy of this resource:

Required:	no; optional
Contained by:	Production unit; Area
Can Contain:	one or more Production Units; Units

Details Tab

You can change the name and description of this resource.

Attribute	Value	Description
Name	Required; user-defined up to 50 characters	Specifies a display name for a resource.
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

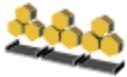
Properties Tab

For more information see **Equipment Properties**.

Classes Tab

For more information, see **Equipment Classes**.

Production Lines



A production line resource represents a group of physical equipment items that are applied sequentially to manufacture a product.

Production lines are applied to discrete production processes. Although smaller equipment items may comprise a production line, they are not defined in the equipment model.

A production line can be the last resource in a hierarchy; that is, a terminating resource. Although work cell resources can be nested within a production line, such nesting indicates that some alternate paths are available within the production line.

The following information describes the hierarchy of this resource:

Required:	no; optional
Contained by:	Area ; Production line
Can Contain:	one or more Production lines; Work cells

Details Tab

You can change the name and description of this resource.

Attribute	Value	Description
Name	Required; user-defined up to 50 characters	Specifies a display name for a resource.
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

Properties Tab

For more information see **Equipment Properties**.

Classes Tab

For more information, see **Equipment Classes**.

Process Cells



A process cell resource represents a group of physical equipment items that is dedicated to a batch manufacturing subprocess. It performs operations on a raw material to produce an interim or final product.

A process cell can be both a terminating and intermediate resource, meaning that a process cell can contain itself or terminate at itself in the equipment hierarchy. A process cell can contain unit resources but only when those units represent alternate production paths through a process cell.

The following information describes the hierarchy of this resource:

Required:	no; optional
Contained by:	Area ; Process cell
Can Contain:	one or more Units ; Process cells

Details Tab

You can change the name and description of this resource.

Attribute	Value	Description
Name	Required; user-defined up to 50 characters	Specifies a display name for a resource.
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

Properties Tab

For more information see **Equipment Properties**.

Classes Tab

For more information, see **Equipment Classes**.

Units



A unit resource represents an alternate production path within a process cell.

In the equipment model, a unit is a terminating resource; that is; the last resource in a hierarchy. Unit resources are contained within a process cell only when some variability exists within the process cell regarding the path a product takes. For example, a unit adds value to a process cell by adding a dye or chemical to the overall process.

The following information describes the hierarchy of this resource:

Required:	no; optional
Contained by:	Process cell; Production unit; Unit
Can Contain:	one or more Units; Equipment modules

Details Tab

You can change the name and description of this resource.

Attribute	Value	Description
Name	Required; user-defined up to 50 characters	Specifies a display name for a resource.

Attribute	Value	Description
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

Properties Tab

For more information see **Equipment Properties**.

Classes Tab

For more information, see **Equipment Classes**.

Work Cells



A work cell resource represents a group of physical equipment items that are applied sequentially to manufacture a product. Work cells most often represent one of multiple alternate paths that can be taken by a product on a production line. Although smaller equipment items may be contained by a work cell, they are not defined in the equipment model.

A work cell is a terminating resource; that is the last resource in a hierarchy. Work cells are included when a production line contains optional production paths.

The following information describes the hierarchy of this resource:

Required:	no; optional
Contained by:	Production lines; Work cells
Can Contain:	one or more Work cells; Equipment modules

Details Tab

You can change the name and description of this resource.

Attribute	Value	Description
Name	Required; user-defined up to 50 characters	Specifies a display name for a resource.

Attribute	Value	Description
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

Properties Tab

For more information see **Equipment Properties**.

Classes Tab

For more information, see **Equipment Classes**.

Storage Zones



A storage zone resource represents a complete inventory subsystem, which has the capability to receive, store, retrieve, move, and ship materials required for production, including material movements between enterprises.

A storage zone can be a terminating resource or can contain storage unit resources if a more detailed inventory structure is required.

The following information describes the hierarchy of this resource:

Required:	no; optional
Contained by:	Areas ; Storage zones
Can Contain:	one or more Storage units ; Storage zones

Details Tab

You can change the name and description of this resource.

Attribute	Value	Description
Name	Required; user-defined up to 50 characters	Specifies a display name for a resource.

Attribute	Value	Description
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

Properties Tab

For more information see **Equipment Properties**.

Classes Tab

For more information, see **Equipment Classes**.

Storage Units



A storage unit resource represents an inventory of a specific material, a group of materials, or a specific type of storage.

The physical location of a storage unit can be dynamic, such as when a specific material inventory is en route to a new location. A storage unit is a terminating resource; that is, the last resource in a hierarchy, and is generally applied when detailed inventory organization is needed.

The following information describes the hierarchy of this resource:

Required:	no; optional
Contained by:	Storage zones; Storage units
Can Contain:	one or more Storage units; Equipment modules

Details Tab

You can change the name and description of this resource.

Attribute	Value	Description
Name	Required; user-defined up to 50 characters	Specifies a display name for a resource.

Attribute	Value	Description
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

Properties Tab

For more information see **Equipment Properties**.

Classes Tab

For more information, see **Equipment Classes**.

Equipment Modules



An equipment module can carry out a finite number of specific, minor processing activities. It combines all necessary physical processing and control equipment to perform those activities.

Equipment modules are usually centered on a piece of processing equipment such as a filter (in batch operations), packager, or an injection molder. Functionally, the scope of the equipment module is defined by the finite tasks it is designed to carry out. .

Physically, an equipment module can be made up of control modules and subordinate equipment modules. For example, a wave soldering machine may be made up of equipment modules for flux application, wave generation, cleaning, or heating, plus a control module for managing the movement of a board through the various equipment modules. A wave generation equipment module might consist of another equipment module for controlling temperature, plus a control module for adjusting wave height

Equipment modules are controlled procedurally through complex logic. They may have multiple states and run through sequences of operations.

Equipment modules are connected to other equipment and systems through control modules.

An equipment module can be a terminating resource; that is, the last resource in a hierarchy.

The following information describes the hierarchy of this resource:

Required:	no; optional
Contained by:	Equipment module; Unit; Storage unit; Work cell

Can Contain:

one or more **control module**; Equipment modules

Details Tab

You can change the name and description of this resource.

Attribute	Value	Description
Name	Required; user-defined up to 50 characters	Specifies a display name for a resource.
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

Properties Tab

For more information see **Equipment Properties**.

Classes Tab

For more information, see **Equipment Classes**.

Control Modules



A control module can consist of a collection of sensors, actuators, other control modules, and associated processing equipment that is operated as a single entity.

A control module is where direct connections to sensors and actuators are made in the factory. Most of the data connections you make to resources are made to control modules. Control modules may be given commands such as start and stop or “set speed”, or can provide information, such as temperature, speed, or alarm information.

A control module can also be made up of other control modules. For example, the heating system in a wave soldering machine may consist of an upper heating coil and a lower heating coil, along with a common temperature output of the solder. You may want to model this as a control module, and then model each heating coil independently as control modules within it. This allows you to control and measure both as a group, or to individually control or monitor them.

Where equipment modules are controlled procedurally through defined sequences or recipes, control modules are driven to a certain state; for example, a temperature element that has a temperature control.

Setting the temperature drives the heating element until the requested temperature is reached. Even if the control module internally has complex logic to attain the state, the exposed interface to the world is simple – set a value, read a value.

A control module can be a terminating resource; that is, the last resource in a hierarchy.

The following information describes the hierarchy of this resource:

Required:	no; optional
Contained by:	Equipment module; Control module.
Can Contain:	one or more Control modules.

Equipment Instance Properties

You can configure values for equipment property attributes. All properties are literal unless the linked option is selected.

In the Properties tab, you must click once in a field to select it, and then click again to edit the field. Depending on the selected field, you can enter text, select from a drop-down list, or click an ellipsis button to open a dialog box.

When you select an equipment class in the Type field, the properties of that class are added as sub-properties to the current property. You can add a class multiple times, which in turn adds multiple sets of that class' properties.

Property Attribute Overrides

If a property is inherited from a class, the default behavior is for its attributes to be linked to the attributes of the property on the parent class. However, you can override the default values of the following attributes:

- Linked (this attribute is available only on equipment instance properties)
- Value
- UoM (Unit of Measure)
- Description

You can override the default values of a property inherited from an equipment class whether the class is associated with an equipment instance, or if a property (on an equipment instance or class) is configured as a class type.

If you override the default value of an attribute on an inherited class property, the new value appears in blue. The blue color indicates that the attribute value is no longer linked to the default value obtained from the property on the parent class.

EngineClass is an equipment class that has a property called RotationsPerMinute. The property type of RotationsPerMinute is RotationsPerMinuteClass. RotationsPerMinuteClass is an equipment class with three properties: Current, Maximum, and Minimum, all of which are inherited by EngineClass as subproperties of RotationsPerMinute.

An equipment instance named Aircraft1 has a property named Engine, which is of type EngineClass. The fully qualified names of all the properties on Aircraft1 are:

- Engine.RotationsPerMinute.Current
- Engine.RotationsPerMinute.Maximum
- Engine.RotationsPerMinute.Minimum

If you override the default attributes of the properties on EngineClass (on the EngineClass property grid) then those changes appear in blue. The overridden values will also be reflected when you view the property grid for Aircraft1, or any other instance or class that inherits from EngineClass; however, they do not appear in blue because they are linked values of the attributes on EngineClass. The values of the property attributes on the RotationPerMinuteClass do not change.

Duplicate Property Attribute Overrides

If you duplicate an equipment instance or class that has overridden values, the overridden state is also duplicated; that is, any overridden attribute values are also duplicated and appear in blue.

Import/Export Property Attribute Overrides

If you export an equipment instance or class that has any overridden attribute values, the value overrides are indicated on the Equipment Property Value tab of the export file as True. You can change this value in the export file to False, and when the file is imported the overridden attribute value returns to the original, default value. If the override value is False, the value in the corresponding column is ignored. For example, if the value in the DescriptionOverridden column is False, the value in the Description column for that property is ignored. In addition, if a property is linked to an external data source, and the value in the ValueOverridden column is changed to False, the data item link on the Property Data Items tab is ignored.

**Note:**

Properties on an equipment instance or class (that is, non-inherited properties added directly to an equipment resource) do not display True or False for value overrides on the export file.

The following table describes the information available on the Properties tab, where you can configure the values for the property attributes.

Attribute	Value	Description
Add Property	User-defined	Click to add a new property.
Delete Property	User-defined	Click to delete the selected property.
Configure Columns	System-defined	<p>Click to open the Configure Columns dialog box where you can select the property columns to show or hide.</p> <div data-bbox="1024 968 1419 1283" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note: The Configure Columns dialog box changes the visible columns for all equipment resource properties.</p> </div>
Name	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 50 characters	Specifies a display name for the selected property.
Type	System-defined	<p>Specifies the <i>data type</i> of the property.</p> <div data-bbox="1024 1570 1419 1843" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note:</p> <ul style="list-style-type: none"> • This property cannot be changed if it is inherited from another resource. </div>

Attribute	Value	Description
		 <ul style="list-style-type: none"> • When you select Browse for class, the Universal Browser appears where you can select an equipment class. • This field is not available if the Linked check box is selected.
Linked	User-defined	<p>Select this check box to bind the property to an external data source. For more information, see <i>Data Sources</i>.</p>  <p>Note: This check box is not available for the property if the property type is class; however, it is available for the properties inherited from a class.</p>
Value	User-defined	<p>Specifies the value for the selected data type. Valid values depend on the selected data type. This field is not available for the property if the property type is class; however, it is available for the sub-properties inherited from the class.</p>

Attribute	Value	Description
		<div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note:</p> <ul style="list-style-type: none"> The method of entering or selecting a value depends on the data type selected in the Type field and whether the Linked check box is selected. If the Linked option is selected, click twice in this field, and then click the ellipsis button to open the Configure Data Source dialog box, where you can select the data source to read from. </div>
UoM (Unit of Measure)	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 20 characters	Specifies the unit of measure for the data type, such as gallons, inches, tons.
Equipment Class	User-defined	Specifies the equipment class associated with the equipment instance. The equipment class properties are inherited by the equipment instance.

Attribute	Value	Description
Description	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 255 characters	Provides more information that a user may find helpful.
Show/Hide Property Details  	N/A	Click to show or hide the property details pane that displays the name, qualified name (for referencing the property), and attributes of a selected property. The Test external data link option is available when a property is linked to an external data source.
Text external data link	N/A	Click to test whether the selected data source is valid. This link is available only when the selected property is linked to an external data source.

Related reference

[Material Lots and Sublots \(on page 416\)](#)

[Material Inventory \(on page 417\)](#)

[Equipment Actuals \(on page 453\)](#)

Equipment Classes

Equipment classes allow you to organize properties into common collections to make the modeling process easier for equipment resources.

You may have equipment that have common properties that you want to apply to a number of equipment resources. For example, every machine that draws power may have a ‘current draw’ property for monitoring energy usage in your plant. Rather than adding this property to every equipment resource, you can create a “Powered Machine” class that contains the common properties for current draw, voltage, and so on. After the class is defined, you can add this class to every equipment resource that uses power. For every resource you add the class to, the properties defined in the class become available so you can read from or write to those properties. This not only makes the process of creating your models easier, it ensures that you have common data across machines for reporting and analysis.

**Important:**

If you change a property within a class, that change affects all resources that the class was added to.

Details Tab

You can change the name and description of this resource.

Attribute	Value	Description
Name	Required; user-defined up to 50 characters	Specifies a display name for a resource.
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

Properties

On the Properties tab of an equipment class, the available columns are Name, Type, Value, Unit of Measure, and Description. There are no Linked or Equipment Class columns.

For more information, see Equipment Class Properties.

Equipment Class Properties

Classes allow you to define standard properties to apply to the equipment resources that make up your production operation.

In the Properties tab, you must click once in a field to select it, and then click again to edit the field. Depending on the selected field, you can enter text, select from a drop-down list, or click an ellipsis button to open a dialog box.

When you select an equipment class in the Type field, the properties of that class are added as sub-properties to the current property. You can add a class multiple times, which in turn adds multiple sets of that class' properties.

Property Attribute Overrides

If a property is inherited from a class, the default behavior is for its attributes to be linked to the attributes of the property on the parent class. However, you can override the default values of the following attributes:

- Linked (this attribute is available only on equipment instance properties)
- Value
- UoM (Unit of Measure)
- Description

You can override the default values of a property inherited from an equipment class whether the class is associated with an equipment instance, or if a property (on an equipment instance or class) is configured as a class type.

If you override the default value of an attribute on an inherited class property, the new value appears in blue. The blue color indicates that the attribute value is no longer linked to the default value obtained from the property on the parent class.

EngineClass is an equipment class that has a property called RotationsPerMinute. The property type of RotationsPerMinute is RotationsPerMinuteClass. RotationsPerMinuteClass is an equipment class with three properties: Current, Maximum, and Minimum, all of which are inherited by EngineClass as subproperties of RotationsPerMinute.

An equipment instance named Aircraft1 has a property named Engine, which is of type EngineClass. The fully qualified names of all the properties on Aircraft1 are:

- Engine.RotationsPerMinute.Current
- Engine.RotationsPerMinute.Maximum
- Engine.RotationsPerMinute.Minimum

If you override the default attributes of the properties on EngineClass (on the EngineClass property grid) then those changes appear in blue. The overridden values will also be reflected when you view the property grid for Aircraft1, or any other instance or class that inherits from EngineClass; however, they do not appear in blue because they are linked values of the attributes on EngineClass. The values of the property attributes on the RotationPerMinuteClass do not change.

Duplicate Property Attribute Overrides

If you duplicate an equipment instance or class that has overridden values, the overridden state is also duplicated; that is, any overridden attribute values are also duplicated and appear in blue.

Import/Export Property Attribute Overrides

If you export an equipment instance or class that has any overridden attribute values, the value overrides are indicated on the Equipment Property Value tab of the export file as True. You can change this value in the export file to False, and when the file is imported the overridden attribute value returns to the original,

default value. If the override value is False, the value in the corresponding column is ignored. For example, if the value in the DescriptionOverridden column is False, the value in the Description column for that property is ignored. In addition, if a property is linked to an external data source, and the value in the ValueOverridden column is changed to False, the data item link on the Property Data Items tab is ignored.

**Note:**

Properties on an equipment instance or class (that is, non-inherited properties added directly to an equipment resource) do not display True or False for value overrides on the export file.

The following table describes the information available on the Properties tab, where you can configure the values for the property attributes.

Attribute	Value	Description
Add Property	User-defined	Click to add a new property.
Delete Property	User-defined	Click to delete the selected property.
Configure Columns	System-defined	Click to open the Configure Columns dialog box where you can select the property columns to show or hide. <div data-bbox="1036 1192 1086 1245" data-label="Image"> </div> <div data-bbox="1094 1201 1175 1234" data-label="Section-Header">Note:</div> <div data-bbox="1094 1245 1391 1461" data-label="Text"> <p>The Configure Columns dialog box changes the visible columns for all equipment resource properties.</p> </div>
Name	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 50 characters	Specifies a display name for the selected property.
Type	System-defined	Specifies the <i>data type</i> of the property.

Attribute	Value	Description
		<div data-bbox="1024 268 1417 1096" style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note:</p> <ul style="list-style-type: none"> This property cannot be changed if it is inherited from another resource. When you select Browse for class, the Universal Browser appears where you can select an equipment class. This field is not available if the Linked check box is selected. </div>
Linked	User-defined	<p>Select this check box to bind the property to an external data source. For more information, see <i>Data Sources</i>.</p> <div data-bbox="1024 1318 1417 1680" style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note:</p> <p>This check box is not available for the property if the property type is class; however, it is available for the properties inherited from a class.</p> </div>
Value	User-defined	<p>Specifies the value for the selected data type. Valid values depend on the selected data type. This field is not available for the prop-</p>

Attribute	Value	Description
		<p>erty if the property type is class; however, it is available for the sub-properties inherited from the class.</p> <div data-bbox="1024 470 1419 1570" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note:</p> <ul style="list-style-type: none"> The method of entering or selecting a value depends on the data type selected in the Type field and whether the Linked check box is selected. If the Linked option is selected, click twice in this field, and then click the ellipsis button to open the Configure Data Source dialog box, where you can select the data source to read from. </div>
UoM (Unit of Measure)	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 20 characters	Specifies the unit of measure for the data type, such as gallons, inches, tons.
Equipment Class	User-defined	Specifies the equipment class associated with the equipment instance. The equipment class

Attribute	Value	Description
		properties are inherited by the equipment instance.
Description	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 255 characters	Provides more information that a user may find helpful.
Show/Hide Property Details  	N/A	Click to show or hide the property details pane that displays the name, qualified name (for referencing the property), and attributes of a selected property. The Test external data link option is available when a property is linked to an external data source.
Text external data link	N/A	Click to test whether the selected data source is valid. This link is available only when the selected property is linked to an external data source.

Related reference

[Equipment Instance Properties \(on page 442\)](#)

[Equipment Classes \(on page 447\)](#)

Related information

[Equipment Model \(on page 421\)](#)

Equipment Actuals

Equipment actuals identify an equipment capability used during a selected work process segment.

Resource actuals show the as-performed actual values for a resource during a production cycle. A resource actual is associated with a resource specification, making it easy to compare actual and specified values for properties.

When the planned use of equipment resources does not match the actual use of equipment resources, then an equipment actual is used. The equipment actual records the true values of resources during

production, which can be compared to expected use. Work process segments have specifications, but no way to record actual values. Using equipment actuals makes it possible to take a true measurement of the resources that were used. For example, if the production line that is normally used is shut down for maintenance and a separate production line is used, then an equipment actual would record the use of the other line.

An equipment actual corresponds to an existing equipment resource.

IProductionRuntime Service Interface

Creating equipment actuals and editing their properties is managed using the **CreateEquipmentActual** and **AddResourceActualProperties** call methods.

Production Model

Production Model

The production model presents a set of resources used to describe a product as a collection of components and the processes applied to those components.

The resource hierarchy provides a modular and layered approach to specifying a variable family of products and their production.

The production model describes activities, such as processes, recipes for making things, and information flows. This model determines how and when the entities described in the other models are used. The production model is also used to describe products that can be produced, as well as the resources (such as equipment and materials), production steps, and processes required to produce them.

The production model is made up of the following elements:

- The static data models that describe resources and processes, such as process segments, definition segments, and data classes.
- The data used to associate resources and processes with an output product, as created with work definitions.
- The runtime data that describes work in progress, including work requests.

Work process and definition segments can be created in a hierarchical structure that allows you to create segments at a very granular level, depending on your business requirements.

Related reference

[Work Requests View \(on page 513\)](#)

[Work Process Segments \(on page 461\)](#)

[Work Requests \(on page 498\)](#)

[Work Definitions \(on page 486\)](#)

[Work Definition Segments \(on page 477\)](#)

[Work Data Classes \(on page 457\)](#)

Add properties or parameters to a resource

Procedure

1. From the navigator, select the model that contains the resource you want to work with.
In the next pane, a list of resources appears.
2. Select the resource you want to work with.
3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click the **Properties** or **Parameters** tab, and then click **Add Property** or **Add Parameter**.
5. In the **Name** and **Description** fields, enter a name and description for the property/parameter.
6. From the **Data Type** list, select a data type.
7. In the **Value** field, enter a value for the selected data type.
8. **Optional:** If applicable, in the **Unit of Measure** field, enter a unit of measure for the selected data type.
9. Click **Save**.

Modify the name and description of a property or parameter

Procedure

1. From the navigator, select the model that contains the resource you want to work with.
In the next pane, a list of resources appears.
2. Select the resource you want to work with.
3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click the **Properties** or **Parameters** tab, and then select the property or parameter you want to modify.
5. In the **Name** and **Description** fields, enter a name and description for the property/parameter.
6. Click **Save**.

Modify the values for a property or parameter

Before you begin

**Important:**

Changing property/parameter values affects your production process.

Procedure

1. From the navigator, select the model that contains the resource you want to work with.
In the next pane, a list of resources appears.
2. From this list, select the resource you want to work with.
3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click the **Properties** or **Parameters** tab, and then select the property or parameter you want to modify.
5. Change the property/parameter's attributes, as required. For more information, see **Add properties to a resource**.
6. In the **Value** field, enter a value for the selected data type.
7. If applicable, in the **Unit of Measure** field, enter a unit of measure for the selected data type.
8. Click **Save**.

Related information

[Add properties to a resource \(on page 427\)](#)

Delete properties or parameters from a resource

Before you begin

**Note:**

When a property/parameter is deleted, it cannot be recovered.

Procedure

- 1.
2. From the navigator, select the model that contains the resource you want to work with.
In the next pane, a list of resources appears.
3. From this list, select the resource you want to work with.
4. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.

5. Click the **Properties** or **Parameters** tab, and then select the property or parameter you want to delete.
6. Click **Delete Property** or **Delete Selection**, and then click **Save**.

Work Data Classes

Work data classes are defined by users and associated with process segments.

When this association occurs, production properties become available to workflows through segment responses. The work data class outlines the data returned by the work process segment.

You can:

- define collections of work data
- associate data classes with work process segments through a workflow's **Call Method activity**
- access this data from the reporting database

Related reference

[Work Requests View \(on page 513\)](#)

[Work Process Segments \(on page 461\)](#)

[Work Requests \(on page 498\)](#)

[Work Definitions \(on page 486\)](#)

[Work Definition Segments \(on page 477\)](#)

Related information

[Production Model \(on page 454\)](#)

Write work data for a work segment response using a workflow

About this task

You can create expressions or bind properties for many of these properties, depending on your workflow requirements.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow. See **Add a Resource to the Workflow Model**.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.

5. Insert an activity:

- At the workflow level:
 - From the Standard Activities panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

6. On the activity configuration panel, click **Configure Method**.

The **Configure Method** dialog box appears, displaying a list of the available call method service providers.

7. Double-click **IProductionRuntime**.**Note:**

If you selected the **Required Class** check box on the **Work Data** tab of the associated work process segment, then go to step 8. If not, go to step 9.

8. Click **AddWorkDataClassToResponse**, and then click **OK**.

- a. In the **Segment Response ID** field, click the arrow, and then select **Bind**.
The **Select Variable** dialog box appears.
- b. From the **Activities** tab, select a **ReturnValue** for the segment response, and then click **OK**.
- c. If required, in **Work Data Class**, bind a property the resource.

9. Click **WriteWorkData**.

- a. If required, to configure the data properties to write, click **Next**.
- b. Click **Add classes of work data properties**, and then select a work data class.
The properties for the selected work data class appear.
- c. To change properties, click **Remove classes of work data properties**, and then select a different work data class.

- d. Click **Select all** or select the check box of a specific property, and then click **Finish**. Each property you select appears as an input parameter on the activity configuration panel.
 - e. For each property, enter a new value or expression, or bind a property or parameter to the method.
 - f. In the **Segment Response ID** field, click the arrow, and then select **Bind**.
The **Select Variable** dialog box appears.
 - g. From the **Activities** tab, select a **ReturnValue** for the segment response, and then click **OK**.
10. Click **Save**.
 11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

Work Data Class Details Tab

The **Details** tab offers general information about the selected work data class. Work data classes are associated with process segments.

You can change the name and description of this resource.

Attribute	Value	Description
Name	Required; user-defined up to 50 characters	Specifies a display name for a resource.
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

Work Data Class Properties Tab

The **Properties** tab allows you to add, modify, and delete properties that are specific to the selected work data class.



Important:

All relevant production resources, work process segments, work data classes with properties, and work requests must exist in order for a value from one workflow to be recorded in another workflow.

The following table describes the attributes specific to this resource type.

Attribute	Value	Description
List	User-defined	Displays a list of properties for the selected resource that can be arranged by name, type, and so on, depending on the property. These properties can be inherited from another resource or created for the current resource.
Add Property	N/A	Click to add a new property.
Delete Property	N/A	Click to delete the selected property.
Name	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 50 characters	Specifies a display name for the selected property.
Description	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 255 characters	Provides more information about a property that a user may find helpful.
Data Type	System-defined	Specifies the data type of the property.  Note: This property cannot be changed if it is inherited from another resource.
Value	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 8000 characters	Specifies the value for the selected data type. Valid values depend on the selected data type.
Unit of Measure	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 20 characters	Specifies the unit of measure for the data type, such as gallons, inches, tons.

Work Process Segments

Work process segments define an activity and the resources required to carry out the activity. It outlines the data associated with a process.

For example, making a bicycle frame requires an assembly jig, a bending machine, and an assembler for 30 minutes.

A work process segment can be used to produce different products depending on how the process is used. In the example described above, the process segment could be used to produce bike frames of different sizes and colors.

Work process segments are made up of material, equipment, and personnel specifications, as well as process parameters that customize the process behavior. The specifications can refer to definitions or classes from a corresponding resource model. You can track resource actual properties that are based on a specification's properties when the work process segment is associated with a segment response. Each specification can define a template for resource actual properties based on the specified resource's properties. A template is used when resource actuals associated with a specification are created.

Resource actual properties are tracked in a segment response when the appropriate call methods (that is, `CreateMaterialActual`, `CreateEquipmentActual`, or `CreatePersonnelActual`) are used in a workflow that is configured to create a segment response. However, for a template to be applied, the `RequirementID` parameter must be used to provide the segment requirement ID so that the specification with the template can be determined.

Alternatively, rather than using the response tracking capability on the Specifications tab to predefine the resource properties to track, you can use the `AddResourceActualProperties` call method to explicitly add resource actual properties to a resource actual to track the resource properties on the associated segment response.

A segment-workflow association defines the associated workflow at design time, then looks it up at run time, and executes it without the need to hard-code the workflow address into another workflow. With this additional flexibility, a workflow could load a work process segment and execute its associated workflow without you having to know which workflows are actually associated with the segment.

A master segment is a work process segment that contains information that applies to all work definition segments associated with a work definition. In other words, a master segment is a template for a work request. For example, a customer name can be represented as a segment parameter in a master segment.

A master segment can be associated with multiple work definitions; however, a work definition can be associated with only one master segment. Work requests are created from an individual work definition and, therefore, can also be associated with only one master segment.

Related reference

[Work Definitions \(on page 486\)](#)

[Work Definition Segments \(on page 477\)](#)

[Actuals \(on page 557\)](#)

Related information

[Production Model \(on page 454\)](#)

Create a work segment response for a work request using a workflow

Before you begin

In a workflow subprocess, you use two call method activities to perform this task. You can create expressions or bind properties for many of these properties, depending on your workflow requirements.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow. See **Add a Resource to the Workflow Model**.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.

The **Configure Method** dialog box appears, displaying a list of the available call method service providers.

7. Double-click **IProductionRuntime** to display the list of available call methods.
8. Click **CreateWorkResponse**.
9.
 - a. Click **Next** to select a work type, and then click **Finish**.
The input parameters appear in the activity configuration panel.
 - b. In the **Description** field, enter a description or expression, or bind a property or parameter to the method.
 - c. In the **Location** field, click the ellipsis button, and then select an equipment location.
 - d. In the **S95 ID** field, enter a name or expression, or bind a property or parameter to the method.
10. In the **Work Request** field, click the ellipsis button.
The **Universal Browser** appears.
11. Select the work request you want to associate with the work response.
12. Repeat steps 5-7.
13. Click **CreateSegmentResponse**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
14. Optionally:
 - a. In the **Description** field, enter a description or expression, or bind a property or parameter to the method.
 - b. In the **End Time** and **Start Time** fields, select a date and time, or add an expression, or bind a property or parameter to the method.
 - c. In the **Location** field, click the ellipsis button, and then select an equipment location.
 - d. In the **S95 ID** field, enter a name or expression, or bind a property or parameter to the method.
15. In the **Work Process Segment** field, click the ellipsis button, and then select a process segment to associate with this segment response.
16. In the **Parent ID** field, click the arrow, and then select **Bind**.
The **Select Variable** dialog box appears.
17. Expand the **CreateWorkResponse** call method added in step 8, select a **ReturnValue** property, and then click **OK**.
18. Click **Save**.

19. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

When the workflow runs, the segment response is created and can be viewed in the Response Information tab of the work request associated with the work request selected in step 11.

Work Process Segment Details Tab

The **Details** tab offers general information about the selected work process segment.

You can change the name and description of this resource.

Attribute	Value	Description
Name	Required; user-defined up to 50 characters	Specifies a display name for a resource.
Description	Required; user-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

The following table describes the attributes specific to this resource type.

Attribute	Value	Description
Location: Change	N/A	Click to open the Universal Browser to select the equipment that you want to use for the process location. It identifies the associated element of the equipment hierarchy model. It may also define the scope of the process segment definition, such as the site or area it is defined for; therefore, it is not intended for locations outside of the scope.
Location: Clear	N/A	Click to clear the equipment location.

Attribute	Value	Description
		 Note: This button is available only if an equipment location is currently set.
Set as Master check box	N/A	Specifies whether the process segment is a master segment.
Process Duration: Duration	User-defined numeric value	Specifies the resource's duration. The duration can be defined at any level of the process. Duration is not inherited through an association with another resource.
Process Duration: Unit of Measure	User-defined alphanumeric string (A-Z, a-z, 0 - 9) up to 20 characters	Specifies the duration's unit of measure, such as gallons, inches, tons.
Work Type	System-defined	Specifies the work type to describe the category of work that the work process segment is associated with. Valid values are: <ul style="list-style-type: none"> • Production • Maintenance • Quality • Inventory • Mixed • Unspecified
Workflow: Change	N/A	Click to open the Universal Browser to select the workflow to associate with this process segment. It identifies the associated element of the workflow hierarchy model.
Workflow: Clear	N/A	Click to clear the workflow association.

Attribute	Value	Description
		 Note: This button is available only if a workflow association is currently set.

Change the equipment location for a process segment

Procedure

1. In the navigator, click **Production**.
2. In the **Folders** list, select the resource type you want to work with.



Tip:

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. On the **Details** tab, in the **Location** section, click **Change**.
The **Universal Browser** appears.
5. Locate and select the equipment resource you want to use, and then click **OK**.



Tip:

To remove the equipment location, click **Clear**.

6. Click **Save**.

Set the duration for a production resource

Procedure

1. In the navigator, click **Production**.
2. In the **Folders** list, select the resource type you want to work with.



Tip:

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. On the **Details** tab, in the **Duration** field, enter the duration.
5. Click **Save**.

Set the unit of measure for a production resource

Procedure

1. In the navigator, click **Production**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. On the **Details** tab, in the **Unit of Measure** field, enter a unit of measure.
5. Click **Save**.

Set the work type for a resource

Procedure

1. In the navigator, click **Production**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. On the **Details** tab, from the **Work Type** drop-down list, select a category of work.
5. Click **Save**.

Set a work definition segment as a master segment

Procedure

1. In the navigator, click **Production**.
2. In the **Folders** list, select the resource type you want to work with.



Tip:

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, click **S95 Model Editor**.
4. Click the **Details** tab.
5. Select the **Set as Master** check box to set the work process segment as a master segment.
6. Click **Save**.

Associate a workflow with a work process segment

Procedure

1. In the navigator, click **Production**.
2. In the **Folders** list, select the resource type you want to work with.



Tip:

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. On the **Details** tab, in the Workflow section, click **Change**.
The **Universal Browser** appears.
5. Locate and select the workflow you want to associate with the work process segment, and then click **OK**.



Note:

To cancel the workflow association, click **Clear**.

6. Click **Save**.

Work Process Segment Parameters Tab

The **Parameters** tab allows you to add, modify, and delete parameters that are specific to the selected work process segment.

Parameters in a work process segment are displayed with any associated work definition segment, where you can override the value.

If you add, delete, or modify a work process segment parameter, any associated work definition segments are also affected.

The following table describes the attributes specific to this resource type.

Attribute	Value	Description
List	User-defined	Displays a list of parameters for the selected resource that can be arranged by name, type, and so on, depending on the parameter. These parameters can be inherited from another resource or created for the current resource.
Add Parameter	N/A	Click to add a new parameter.
Delete Selection	N/A	Click to remove the selected parameter.
Name	User-defined alphanumeric string (A-Z, a-z, 0 - 9) up to 20 characters	Specifies a display name for the selected property.
Description	User-defined alphanumeric string (A-Z, a-z, 0 - 9) up to 20 characters	Provides more information that a user may find helpful.
Data Type	System-defined	Specifies the data type of the parameter. This parameter cannot be changed if it is inherited from another resource.

Attribute	Value	Description
Value	User-defined alphanumeric string (A-Z, a-z, 0 - 9) up to 20 characters	Specifies the value for the selected data type. Valid values depend on the selected data type.
UoM (Unit of Measure)	User-defined alphanumeric string (A-Z, a-z, 0 - 9) up to 20 characters	Specifies the unit of measure for the data type, such as gallons, inches, tons.

Related reference

[Work Process Segments \(on page 461\)](#)

Related information

[Production Model \(on page 454\)](#)

Work Process Segment Specifications Tab

The **Specifications** tab allows you to add, modify, and delete equipment, material, and personnel specifications for the selected work process segment.

The following table describes the attributes specific to this resource type.

Attribute	Value	Description
Specifications list	User-defined	Displays a list of associated equipment, material, and personnel resources that can be sorted by name or type.
Add Equipment	N/A	Click to open the Universal Browser to select the resource that you want to use for the process. <div data-bbox="1024 1591 1419 1816" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Tip: You can modify the quantity and unit of measure.</p> </div>

Attribute	Value	Description
Add Material	N/A	<p>Click to open the Universal Browser to select the resource that you want to use for the process.</p> <div data-bbox="1027 478 1419 743" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Tip: You can select how the material is used, and modify the quantify and unit of measure.</p> </div>
Add Personnel	N/A	<p>Click to open the Universal Browser to select the resource that you want to use for the process.</p> <div data-bbox="1027 970 1419 1192" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Tip: You can modify the quantity and unit of measure.</p> </div>
Delete Selection	N/A	<p>Click to remove the selected specification.</p>
Settings: Material Use	User-defined	<p>Select the option that describes the material specification. Valid values are:</p> <ul style="list-style-type: none"> • Produced • Consumed • Consumable <div data-bbox="1027 1663 1419 1810" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: This options is available only when a material</p> </div>

Attribute	Value	Description
		 specification is configured.
Settings: Quantity	User-defined numeric value	Specifies the required quantity for the resource.
Settings: Unit of Measure	User-defined alphanumeric string (A-Z, a-z, 0 - 9) up to 20 characters	Specifies the unit of measure, such as gallons, inches, tons.
Response Tracking: Property Name	User-defined alphanumeric string (A-Z, a-z, 0 - 9) up to 20 characters	Displays the specification's property name.
Response Tracking: Class Name	User-defined alphanumeric string (A-Z, a-z, 0 - 9) up to 20 characters	Displays the specification's class names, where applicable.
Response Tracking: Respond check box	N/A	<p>Select this check box for each resource property whose resource actual property you want to track on an associated segment response.</p> <div data-bbox="1024 1213 1419 1791" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;"> <p> Tip: If you do not select this check box, you can still track resource properties on a segment response by using the <code>AddResourceActualProperties</code> call method in a workflow to explicitly add resource actual properties to a resource actual.</p> </div>

Related reference[Work Process Segments \(on page 461\)](#)[Actuals \(on page 557\)](#)**Related information**[Production Model \(on page 454\)](#)

Add specifications to a process segment

Before you begin**Note:**

You must create equipment, material, and personnel resources before you can add them as specifications.

Procedure

1. In the navigator, click **Production**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click the **Specifications** tab.
5. If required, add the following resources:

Click...	To add...
Add Equipment	an equipment or location resource
Add Material	a material resource
Add Personnel	people or personnel class resources

The **Universal Browser** appears.

6. Locate and select the resource you want to add as a specification, and then click **OK**.
7. In the **Name** and **Description** fields, enter a name and description for the specification.
8. **Optional:** In the **Quantity** and **Unit of Measure** fields, enter a quantity and unit of measure.

9. For material specifications, select a **Material Use** option.
10. **Optional:** Select the **Response Tracking** check box for each resource property you want to track on a segment response.
11. Click **Save**.

Related reference

[Work Definition Segments \(on page 477\)](#)

[Actuals \(on page 557\)](#)

Modify the settings for a specification

Procedure

1. In the navigator, click **Production**.
2. In the **Folders** list, select the resource type you want to work with.



Tip:

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click the **Specifications** tab, and then select a specification.
5. If applicable, for the selected resource:
 - a. Click **Edit**, and in the **Name** and **Description** fields, enter a new name or description.
 - b. For a material resource, select a **Material Use** option.
 - c. In the **Quantity** field, enter a new quantity.
 - d. In the **Unit of Measure** field, enter a new unit of measure.
6. Click **Save**.

Delete a specification from a process segment

Procedure

1. In the navigator, click **Production**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.

The editor appears in the workspace, displaying the resource information.

4. Click the **Specifications** tab, and then select the specification you want to delete.
5. Click **Delete Selection**, and then click **Save**.

**Note:**

When a resource is deleted, it cannot be recovered.

Work Process Segment Work Data Tab

The **Work Data** tab for the selected work process segment allows you to add, modify, and delete work data classes.

Work data classes associated with a work process segment are inherited by a work definition segment when it is associated with the work process segment. Each class adds its collection of properties to the work data associated with the work process segment.

For more information, see **Work Data Classes**.

The following table describes the attributes specific to this resource type.

Attribute	Value	Description
Class List	User-defined	Displays a list of work data classes that can be sorted by name.
Add Class	N/A	Click to open the Universal Browser to select the work data class that you want to use for the process segment.



Note:
You can use up to 50 work data classes per work process segment.

Attribute	Value	Description
		 The options available to you are in bold.
Delete Selection	N/A	Click to remove the selected work data class.
Required Class	User-defined	Select this check box to ensure that the work data for a work process segment is automatically applied to segment responses.
Properties	User-defined	Displays all the properties configured on the selected work data class.

Related reference

[Work Process Segments \(on page 461\)](#)

[Work Data Classes \(on page 457\)](#)

Add a work data class to a process segment

Before you begin**Note:**

You must have created work data classes before you can add them. For more information, see **Add a resource to a model**.

Procedure

1. In the navigator, click **Production**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.

4. Click the **Work Data** tab.
5. Click **Add Class**.
The **Universal Browser** appears.
6. Select the work data class you want to add, and then click **OK**.

**Note:**

You can add up to 5 data classes per process segment. The options available to you are in bold print.

**Tip:**

To remove a work data class, select the appropriate resource, and then click **Delete Selection**.

7. If applicable, select the **Required Class** check box.
8. Click **Save**.

Work Definition Segments

Work definition segments reference a single work process segment and refine the details that describe what the work process segment will accomplish.

A definition segment defines how a process segment is used to produce a specific product by feeding parameters and specifications into a work process segment.

**Note:**

A work definition segment can be associated with only one work process segment.

You can also associate work definition segments with the parameters of work process segments through a workflow's **Call Method activity**.

Work definition segments further detail the resources that are required to create the product; that is, the equipment, material, and personnel.

The definition segment inherits the parameters and specifications defined for the associated process segment. You can customize the definition segment by adding product-specific parameters and by changing the values of inherited process segment parameters. However, you cannot delete parameters inherited from the process segment. If you override the value of a process segment parameter at the

definition segment level, the change applies only to the specific definition segment; it is not reflected in the process segment.

Equipment, material, and personnel specifications are also inherited from the associated process segment; you cannot add specifications to the definition segment, nor can you change the name and description of an inherited specification; however, you can change the quantity.

The following topics describe the S95 Editor tabs and available attributes.

Related reference

[Work Requests View \(on page 513\)](#)

[Work Process Segments \(on page 461\)](#)

[Work Requests \(on page 498\)](#)

[Work Definitions \(on page 486\)](#)

[Work Data Classes \(on page 457\)](#)

Related information

[Production Model \(on page 454\)](#)

Segment Hierarchies

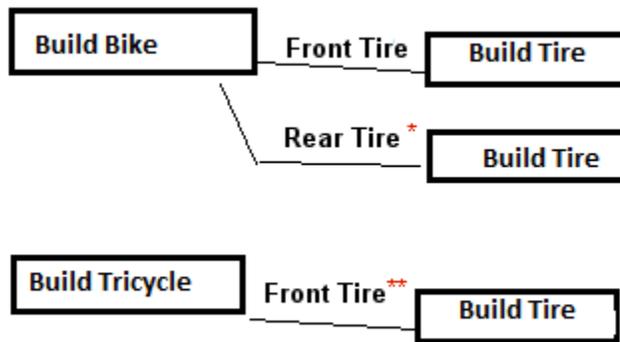
A segment hierarchy is a collection of work definition segments in a parent/child relationship.

You can create a segment hierarchy with a maximum of four levels. A work definition segment can be associated with multiple child segments; however, the association name for each relationship must be unique.

**Important:**

A work definition segment cannot be a child of another work definition segment if it is already a parent to that segment. As well, a work definition segment cannot be a parent to another work definition segment if it is already a child of that segment.

For example, you have two parent segments, named *Build Bicycle* and *Build Tricycle*. For *Build Bicycle*, you have two *Build Tirechild* segments. Even though the child segment being used is the same for both tires, you must provide a unique name to each one of the associations: *Front Tire* and *Rear Tire*. The *Build Tricycle* segment could also have a child segment named *Front Tire*. Even though the name of the association is the same as *Build Bicycle*, the association is unique because it is for a different parent segment.



*You cannot use the name *Front Tire* twice because that name is already being used for another relationship within the *Build Bike* hierarchy.

**Even though the *Build Tricycle* hierarchy is the same as *Build Bike*, you can use the name *Front Tire* because there is no other relationship with that name within the *Build Tricycle* hierarchy.

A parent work definition segment can be a container for its child segments, or it can have its own execution steps. You can set the execution order on all levels of work definition segments within a hierarchy.

Work Definition Segment Details Tab

The **Details** tab offers general information about the selected work definition segment.

You can change the name and description of this resource.

Attribute	Value	Description
Name	Required; user-defined up to 50 characters	Specifies a display name for a resource.
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

The following table describes the attributes specific to this resource type.

Attribute	Value	Description
Related Work Process Segment	User-defined	Specifies the process segment associated with the definition segment.
Work Definition Segment Duration: Duration	User-defined numeric value	Specifies the resource's duration in minutes. The duration can be defined at any level of the process. Duration is not inherited through an association with another resource.
Set as Master Segment	User-defined	Specifies whether the work definition segment is a master segment. This check box is read-only. You can set the master segment on the Work Process Segments Details tab .

Related reference

[Work Process Segment Details Tab \(on page 464\)](#)

Set the duration for a production resource

Procedure

1. In the navigator, click **Production**.
2. In the **Folders** list, select the resource type you want to work with.



Tip:

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. On the **Details** tab, in the **Duration** field, enter the duration.
5. Click **Save**.

Set a work definition segment as a master segment

Procedure

1. In the navigator, click **Production**.
2. In the **Folders** list, select the resource type you want to work with.



Tip:

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, click **S95 Model Editor**.
4. Click the **Details** tab.
5. Select the **Set as Master** check box to set the work process segment as a master segment.
6. Click **Save**.

Work Definition Segment Parameters Tab

The **Parameters** tab allows you to add, modify, and delete parameters that are specific to the selected work definition segment, and to view the parameters for the associated work process segment.

The following table describes the attributes specific to this resource type.

Attribute	Value	Description
List	User-defined	Displays a list of parameters for the selected resource that can be arranged by name, type, and so on, depending on the parameter. These parameters can be inherited from another resource or created for the current resource.
Add Parameter	N/A	Click to add a new parameter.
Delete Selection	N/A	Click to remove the selected parameter.
Name	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 50 characters	Specifies a display name for the selected parameter.

Attribute	Value	Description
Description	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 255 characters	Provides more information about a parameter that a user may find helpful.
Data Type	System-defined	Specifies the data type of the parameter. <div data-bbox="1024 541 1414 804" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: This parameter cannot be changed if it is inherited from another resource. </div>
Value	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 255 characters	Specifies the value for the selected data type. Valid values depend on the selected data type.
Unit of Measure	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 20 characters	Specifies the unit of measure for the data type, such as gallons, inches, tons.

Override a parameter value

Procedure

1. In the navigator, click **Production**.
2. In the **Folders** list, select the resource type you want to work with.



Tip:

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, click **S95 Model Editor**.
4. On the **Parameters** tab, select a work process segment parameter.
5. In the **Value Override** section, click the **Override Value** check box.

**Note:**

To reset the original value, clear the check box.

6. In the **Value** field, enter a new value.
7. Click **Save**.

Work Definition Segement Specifications Tab

The **Specifications** tab for the selected work definition segment displays the equipment, material, and personnel specifications inherited from the associated work process segment.

When process segment specifications are modified, added, or removed, this information is reflected in the specifications tab of the associated work definition segments. However, you can modify select information related to the specification, such as the quantity and a non-class resource.

The following table describes the attributes specific to this resource type.

Attribute	Value	Description
Specifications list	User-defined	Displays a list of associated equipment, material, and personnel resources that can be sorted by name or type.
Select Equipment button	N/A	Click to open the Universal Browser to define this resource more specifically by selecting a non-class resource.
Select Material button	N/A	Click to open the Universal Browser to define this resource more specifically by selecting a non-class resource.
Select Personnel button	N/A	Click to open the Universal Browser to define this resource more specifically by selecting a non-class resource.
Settings: Quantity field	User-defined numeric value	Specifies the required quantity for the resource.

Attribute	Value	Description
Clear button	User-defined	Click to remove the newly selected resource.

Modify a specification in work definition segments

Procedure

1. In the navigator, click **Production**.
2. In the **Folders** list, select the resource type you want to work with.



Tip:

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, click **S95 Model Editor**.
4. Click the **Specifications** tab, and then select a specification.
5. If applicable, click **Select** to select a new resource.
 - Using the Universal Browser, select a resource, and then click **OK**.
 - Click **Clear** to remove the selection.
6. In the **Quantity** field, enter a new amount.
7. Click **Save**.

Work Definition Segment Work Data Tab

The **Work Data** tab for the selected work definition segment allows you to view all work data classes that are inherited from an associated work process segment.

For more information, see [Work Data Classes \(on page 457\)](#).

Work Definition Segment Child Segments Tab

The **Child Segments** tab allows you to associate one or more child work definition segments with the selected parent work definition segment to create a hierarchy of work definition segments.

The following table describes the attributes specific to this resource type.

Attribute	Value	Description
Segment Order list	User-defined	Displays a list of associated child work definition segments

Attribute	Value	Description
		that can be arranged by order or name.
Associate Child Segment	N/A	Click to open the Universal Browser to select the work definition segment that you want to be a child of the current work definition segment.
Remove Child Segment	N/A	Click to remove the selected child work definition segment from the current hierarchy.
Name	User-defined	Specifies a display name for the selected work definition segment.
Order	User-defined numeric value	Specifies the execution order of an associated work definition segment.
Associated Work Definition Segment	User-defined	Displays the child work definition segment associated with the parent work definition segment.
Related Work Process Segment	System-defined	Displays the work process segment associated with the work definition segment.
Segment Hierarchy	System-defined	Displays a collapsed view of the hierarchy for the selected child work definition segment. Expand each level to display the full hierarchy.

Create a segment hierarchy

Before you begin



Important:

A work definition segment cannot be a child of another work definition segment if it is already a parent to that segment. As well, a work definition segment cannot be a parent to another work definition segment if it is already a child of that segment.

Procedure

1. In the Navigator, click **Production > Work Definition Segments**.
2. From the **Work Definition Segments** list, select the resource you want to work with.
3. In the **Displays** panel, click **S95 Model Editor**.
4. On the Child Segments tab, click **Associate child segment**.
The **Universal Browser** appears.
5. From the resource list, select the work definition segment you want to associate with the current work definition segment, and then click **OK**.



Tip:

To remove a segment association, select the appropriate child work definition segment, and then click **Remove child segment**.

6. If applicable, click **Edit**, and then in the **Name** field, enter a name for the segment association.
7. In the **Order** field, enter the execution order for the associated child work definition segment.
8. Click **Save**.

Work Definitions

A work definition is a collection of work definition segments assembled together to outline and define the resources and data required to carry out a unit of work.

A unit of work is a defined task or series of tasks required to carry out an activity. The work definition itself does not describe how to do the work; it describes the specifications and information you need to do it. A work definition is a means to monitor steps in the production process.

By associating work definitions with other production requirements, you ensure that every level of the production process is relevant and functional. Each step within the production model is interrelated, so each entry point allows you to associate resources and processes with one another. Each step must interrelate so that, during production, you can ensure that there are resources (both human and equipment) to produce a product. As such, work definitions are used to generate **work requests**.

Related reference[Work Requests \(on page 498\)](#)**Related information**[Enter a related production rule \(on page 488\)](#)[Set version control \(on page 488\)](#)[Set the work type for a resource \(on page 467\)](#)[Associate a work definition with a work definition segment \(on page 494\)](#)[Create a work request using a workflow \(on page 498\)](#)

Work Definitions Details Tab

The **Details** tab offers general information about the selected work definition.

You can change the name and description of this resource.

Attribute	Value	Description
Name	Required; user-defined up to 50 characters	Specifies a display name for a resource.
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

The following table describes the attributes specific to this resource type.

Attribute	Value	Description
Related Production Rule	User-defined alphanumeric string (A-Z, a-z, 0 - 9) up to 20 characters	Specifies an identification for a rule.
Work Definition Version	User-defined alphanumeric string (A-Z, a-z, 0 - 9) up to 20 characters	Specifies the current version of this resource.
Work Type	System-defined	Specifies the work type to describe the category of work that the work definition is associated with. Valid values are: Produc-

Attribute	Value	Description
		tion, Maintenance, Quality, Inventory, Mixed, and Unspecified.
Linked Resource	System defined	Specifies a linked resource (such as a material definition) that the work definition is associated with. Valid values are: Equipment, Material, and Personnel resources.

Related information

[Link a resource to a work definition \(on page 489\)](#)

Enter a related production rule

Procedure

1. In the navigator, click **Production**.
2. In the **Folders** list, select the resource type you want to work with.



Tip:

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. On the **Details** tab, in the **Production Rule** field, enter a production rule.
5. Click **Save**.

Set version control

Procedure

1. In the navigator, click **Production**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. On the **Details** tab, in the **Version** field, enter a version number.
5. Click **Save**.

Set the work type for a resource

Procedure

1. In the navigator, click **Production**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. On the **Details** tab, from the **Work Type** drop-down list, select a category of work.
5. Click **Save**.

Link a resource to a work definition

Linking a resource to a work definition helps to identify the purpose for that work definition.

Before you begin

For example, linking a material resource to a work definition helps to illustrate that the reason for the work definition is to create the material described in the associated material resource.

Procedure

1. In the navigator, click **Production**.
2. Select Work Definitions, and then select the resource you want to work with.

**Note:**

All areas that are compatible with the current resource appear. Green icons indicate a compatible view.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click the **Details** tab.
5. In the Linked Resource area, click **Change** to select the linked resource from the Universal Browser.
6. Click **OK**, and then click **Save**.

Set a work definition segment as a master segment

Procedure

1. In the navigator, click **Production**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, click **S95 Model Editor**.
4. Click the **Details** tab.
5. Select the **Set as Master** check box to set the work process segment as a master segment.
6. Click **Save**.

Remove a master segment designation

Before you begin

You can change the work process segment data and associations, but if you clear the Set as Master check box for a work process segment that is already being used as a master segment, an error is displayed in the Feedback Zone. To make changes to a master segment, you must first remove it from the associated work definition.

Procedure

1. In the navigator, click **Production**.
2. Select **Work Process Segments** and then select the resource you want to work with.

**Note:**

All areas that are compatible with the current resource appear. Green icons indicate a compatible view.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click the **Details** tab.
5. Clear the **Set as Master** checkbox to remove the master segment designation from the work process segment.
6. Click **Save**.

Master Segment Tab

The **Master Segment** tab allows you to add a master segment (that is, a work process segment set as a master) to the selected work definition.

A master segment can be associated with multiple work definitions; however, a work definition can be associated with only one master segment. Work requests are created from an individual work definition and, therefore, can also be associated with only one master segment.

When a master segment is added to a work definition, its associated details are displayed.

The following table describes the attributes specific to this resource type.

Attribute	Value	Description
Add/Remove a Master Segment	User-defined alphanumeric string (A-Z, a-z, 0 - 9) up to 20 characters	Displays the work definition segment and its associated master segment associated with the selected work definition. <div data-bbox="1024 1472 1419 1740" style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: Only a work definition segment with an associated master segment can be selected. </div>

Attribute	Value	Description
Change	N/A	Click to select the master segment to associate with the work definition.
Clear	N/A	Click to remove the associated master segment from the work definition.
Master Segment Details	User-defined	Displays the parameters and specifications associated with the selected master segment.

Change or add a master segment for a work definition

Before you begin

You can change the work process segment data and associations, but if you clear the *Set as Master* check box for a work process segment that is already being used as a master segment, an error is displayed in the Feedback Zone.

Procedure

1. In the navigator, click **Production**.
2. Select **Work Definitions**, and then select the resource you want to work with.



Note:

All areas that are compatible with the current resource appear. Green icons indicate a compatible view.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click the **Master Segment** tab.
5. Click **Change** to create or change the master segment association.



Note:

You can select only a work definition segment whose associated work process segment is set as a master segment. Valid work definition segments are indicated with bold text in the Universal Browser.

6. Click **OK**. The selected work definition segment, its associated work process segment, and the master segment details are displayed in the Master Segment tab.
7. Click **Save**.

Remove a master segment from a work definition

Before you begin

If you try to change or delete a segment that is already being used as a Master Segment, an error is generated. Before you can make any changes to the master segment, you must first remove it from the work definition.

Procedure

1. In the navigator, click **Production**.
2. Select **Work Definitions**, and then select the resource you want to work with.



Note:

All areas that are compatible with the current resource appear. Green icons indicate a compatible view.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click the **Master Segment** tab.
5. Click **Clear** to remove the master segment from the work definition.
6. Click **Save**.

Work Definition Segments Tab

The **Segments** tab allows you to associate work definition segments with the selected work definition.

You can associate the selected work definition with up to 50 work definition segments. You can also apply an order to the execution of the associated work definition segment.

The following table describes the attributes specific to this resource type.

Attribute	Value	Description
Associated Segments list	User-defined	Displays a list of associated work definition segments that can be arranged by order or name.

Attribute	Value	Description
Associate Segment	N/A	Click to open the Universal Browser to select the work definition segment that you want to associate with this work definition.
Remove Selection	N/A	Click to remove the selected work definition segment from this work definition.
Name	User-defined	Specifies a display name.
Order	User-defined numeric value	Specifies the execution order of an associated work definition segment.
Associated Work Definition Segment	User-defined	Displays the work definition segment associated with the work definition.
Related Work Process Segment	User-defined	Displays the work process segment related to the associated work definition segment.
Segment Hierarchy	User-defined	Displays the master and any associated child segments in the segment hierarchy.

Associate a work definition with a work definition segment

Before you begin



Note:

You can add up to 50 segments to a work definition.

Procedure

1. In the navigator, click **Production**.
2. in the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.

The editor appears in the workspace, displaying the resource information.

4. On the **Segments** tab, click **Associate Segment**.

The **Universal Browser** appears.

5. From the resource list, select the work definition segment you want to associate the work definition with, and then click **OK**.

**Tip:**

To remove a segment association, select the appropriate segment, and then click **Remove Selection**.

6. If applicable, click **Edit**, and then in the **Name** field, enter a name for the segment.

7. In the **Order** field, enter the execution order for the associated work definition segment.

8. Click **Save**.

Work Definition Parameters Tab

The **Parameters** tab allows you to add, modify, and delete parameters that are specific to the selected work definition, and to view parameters that are inherited from an associated work definition segment.

Work process segment parameters are inherited by associated work definition segments. If you add, modify, or delete a work process segment parameter, all associated work definition segments are also affected.

The following table describes the attributes specific to this resource type.

Attribute	Value	Description
List	User-defined	Displays a list of parameters for the selected resource that can be arranged by name, type, and so on, depending on the parameter. These parameters can be inherit-

Attribute	Value	Description
		ed from another resource or created for the current resource.
Add Parameter	N/A	Click to add a new parameter.
Delete Selection	N/A	Click to remove the selected parameter.
Name	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 50 characters	Specifies a display name for the selected parameter.
Description	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 255 characters	Provides more information about a parameter that a user may find helpful.
Data Type	System-defined	Specifies the data type of the parameter. This parameter cannot be changed if it is inherited from another resource.
Value	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 255 characters	Specifies the value for the selected data type. Valid values depend on the selected data type.
UoM (Unit of Measure)	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 255 characters	Specifies the unit of measure for the data type, such as gallons, inches, tons.

Related information

[Production Model \(on page 454\)](#)

Work Definition Specifications Tab

The **Specifications** tab for the selected work definition displays the equipment, material, and personnel specifications inherited from the associated work definition segment.

When work process segment specifications are modified, added, or removed, this information is reflected in the specifications tab of the associated work definition. However, you can modify select information related to the specification, such as the quantity and a non-class resource.

The following table describes the attributes specific to this resource type.

Attribute	Value	Description
Specifications list	User-defined	Displays a list of associated equipment, material, and personnel resources that can be sorted by name or type.
Select Equipment button	N/A	Click to open the Universal Browser to define this resource more specifically by selecting a non-class resource.
Select Material button	N/A	Click to open the Universal Browser to define this resource more specifically by selecting a non-class resource.
Select Personnel button	N/A	Click to open the Universal Browser to define this resource more specifically by selecting a non-class resource.
Settings: Quantity field	User-defined numeric value	Specifies the required quantity for the resource.
Clear button	User-defined	Click to remove the newly selected resource.

For a complete description of the specification attributes of this resource, see **Specifications tab** .

Related reference

[Work Requests View \(on page 513\)](#)

[Work Process Segments \(on page 461\)](#)

[Work Requests \(on page 498\)](#)

[Work Process Segment Specifications Tab \(on page 470\)](#)

Work Definition Work Data Tab

The **Work Data** tab for the selected work definition allows you to view all work data classes that are inherited from an associated work definition segment.

For more information, see **Work Data Classes**.

Related reference

[Work Data Classes \(on page 457\)](#)

Work Requests

Work requests contain the information required to begin a production process.

They are sent by a scheduling system or entered by an operator to trigger an activity that can be individually monitored, scheduled, and reported on. By submitting a work request, the necessary resources are identified and become part of a work schedule. Each work request is associated with a work definition.

Work requests are also associated with the events model. When a work request is created, any scheduled event is triggered in this section. It can be triggered automatically through a service provider from an external system, such as a master scheduler in a plant, or an alarm system, or manually through a user-defined workflow schedule.

Work requests can be created using the:

- **Call Method activity** in the Workflow model; or
- Work Requests resource type in the Production model.

The following topics describe the S95 Editor tabs and available attributes.

Related information

[Production Model \(on page 454\)](#)

Create a work request using a workflow

This procedure describes how to create and configure a work request using a workflow.

About this task

You can create expressions or bind properties for many of these properties, depending on your workflow requirements.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, add a resource to the workflow model.
3. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow. See **Add a resource to the workflow model**.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. Insert an activity:

- At the workflow level:

- From the Standard Activities panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:

- Double-click the local subprocess you want to add the activity to.
- From the Standard Activities panel, click and drag the activity you want to insert to a drop point.

7. On the activity configuration panel, click **Configure Method**.

The Configure Method dialog box appears, displaying a list of the available call method service providers.

8. Double-click **IProductionRuntime** to display the list of available call methods.
9. Click **CreateWorkRequest**, and then click **Finish**.

The input parameters appear in the activity configuration panel.

10. Optionally:

- a. In the **Description** field, enter a description or expression, or bind a property or parameter to the method.
- b. In the **S95 ID** field, enter a name or expression, or bind a property or parameter to the method.

11. In the **Work Definition** field, click the ellipsis button, and then select a work definition.

12. Click **Save**.

13. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

Move work responses to a new work request

You can move a work response from one work request to another work request at any time, regardless of the status of the originating work request.

Procedure

1. Create a new work request or ensure that you have a work request that you can move your work responses to.
2. In the navigator, click **Workflow > Workflows**.
3. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
7. On the activity configuration panel, click **Configure Method**.

The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 8. Double-click **IProductionRuntime** to display the list of available call methods.
 9. Click **RepurposeWorkResponse**, and then click **Finish**.

The input parameters appear in the activity configuration panel.
 10. Configure the call method as follows:
 - a. In the **Destination Work Request** field, click the ellipsis button, and then select the work request whose work responses you want to move.
 - b. In the **Origin Work Request** field, click the ellipsis button, and then select the work request you want to move the work responses to.

11. Click **Save**.
12. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

Create a work segment response for a work request using a workflow

Before you begin

In a workflow subprocess, you use two call method activities to perform this task. You can create expressions or bind properties for many of these properties, depending on your workflow requirements.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow. See **Add a Resource to the Workflow Model**.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.

The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IProductionRuntime** to display the list of available call methods.
 8. Click **CreateWorkResponse**.
 9.
 - a. Click **Next** to select a work type, and then click **Finish**.

The input parameters appear in the activity configuration panel.

- b. In the **Description** field, enter a description or expression, or bind a property or parameter to the method.
 - c. In the **Location** field, click the ellipsis button, and then select an equipment location.
 - d. In the **S95 ID** field, enter a name or expression, or bind a property or parameter to the method.
10. In the **Work Request** field, click the ellipsis button.
The **Universal Browser** appears.
11. Select the work request you want to associate with the work response.
12. Repeat steps 5-7.
13. Click **CreateSegmentResponse**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
14. Optionally:
 - a. In the **Description** field, enter a description or expression, or bind a property or parameter to the method.
 - b. In the **End Time** and **Start Time** fields, select a date and time, or add an expression, or bind a property or parameter to the method.
 - c. In the **Location** field, click the ellipsis button, and then select an equipment location.
 - d. In the **S95 ID** field, enter a name or expression, or bind a property or parameter to the method.
15. In the **Work Process Segment** field, click the ellipsis button, and then select a process segment to associate with this segment response.
16. In the **Parent ID** field, click the arrow, and then select **Bind**.
The **Select Variable** dialog box appears.
17. Expand the CreateWorkResponse call method added in step 8, select a ReturnValue property, and then click **OK**.
18. Click **Save**.
19. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.
When the workflow runs, the segment response is created and can be viewed in the Response Information tab of the work request associated with the work request selected in step 11.

Write work data for a work segment response using a workflow

You can create expressions or bind properties for many of these properties, depending on your workflow requirements.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.

The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IProductionRuntime** to display the list of available call methods.

**Note:**

If you selected the **Required Class** check box on the **Work Data** tab of the associated work process segment, then go to step 8. If not, go to step 9.

8. Click **AddWorkDataClassToResponse**, and then click **OK**.
 - a. In the **Segment Response ID** field, click the arrow, and then select **Bind**.

The **Select Variable** dialog box appears.
 - b. From the **Activities** tab, select a **ReturnValue** for the segment response, and then click **OK**.
 - c. If required, in **Work Data Class**, bind a property the resource.
9. Click **WriteWorkData**.

- a. If required, to configure the data properties to write, click **Next**.
 - b. Click **Add classes of work data properties**, and then select a work data class.
The properties for the selected work data class appear.
 - c. To change properties, click **Remove classes of work data properties**, and then select a different work data class.
 - d. Click **Select all** or select the check box of a specific property, and then click **Finish**. Each property you select appears as an input parameter on the activity configuration panel.
 - e. For each property, enter a new value or expression, or bind a property or parameter to the method.
 - f. In the **Segment Response ID** field, click the arrow, and then select **Bind**.
The **Select Variable** dialog box appears.
 - g. From the **Activities** tab, select a **ReturnValue** for the segment response, and then click **OK**.
10. Click **Save**.
 11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

Work Request Properties

The Details tab offers general information about the selected work request.

You can change the name and description of this resource.

Attribute	Value	Description
Name	Required; user-defined up to 50 characters	Specifies a display name for a resource.
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

The following table describes the attributes specific to this resource type.

Attribute	Value	Description
General Information	User-defined	Displays information about the work request. Information includes the associated work definition and the creation time and work type of the work definition.
Location: Change	N/A	Click to open the Universal Browser to select the equipment that you want to use for this work request. The location is used to provide information about the valid equipment for the selected request.
Location: Clear	N/A	Click to clear the equipment location. <div data-bbox="1024 961 1419 1184" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;">  Note: This button is available only if an equipment location is currently set. </div>
Scheduling: Priority	User-defined numeric value	Specifies the order in which the work request will be performed. This setting can be used with automatic scheduling systems. For manual schedules, set the start and end times. <div data-bbox="1024 1503 1419 1768" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;">  Note: Setting the priority is for internal use only; it does not affect the execution of a work request. </div>
Scheduling: Start Time	User-defined	Specifies the start date and time for the work request. This setting

Attribute	Value	Description
		can be used with priority scheduling.
Scheduling: End Time	User-defined	Specifies the end date and time for the work request. This setting can be used with priority scheduling.
Status	System-defined	<p>Specifies the status of the work request to provide information about a production event and to schedule workflows. Valid values are:</p> <ul style="list-style-type: none"> • Draft • Unscheduled • Pending • Running • Paused • Completed • Failed • Cancelled • Unknown • Finished <div data-bbox="1024 1314 1419 1625" style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: Changing the status can alter another production process. Status changes are signaled in the Events model. </div>

Related reference

[Condition Events \(on page 524\)](#)

Change the equipment location for a process segment

Procedure

1. In the navigator, click **Production**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. On the **Details** tab, in the **Location** section, click **Change**.
The **Universal Browser** appears.
5. Locate and select the equipment resource you want to use, and then click **OK**.

**Tip:**

To remove the equipment location, click **Clear**.

6. Click **Save**.

Schedule a work request

Before you begin

Creating a work request signals a scheduled event in the **events model**.

Procedure

1. In the navigator, click **Production**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. On the **Details** tab, in the **Scheduling** section:

- To schedule an automatic work request, in the **Priority** field, enter a numeric value.
 - To schedule a manual work request, in the **Start Time** and **End Time** fields, select a start and end date and time.
5. Click **Save**.

Related reference

[Condition Events \(on page 524\)](#)

Set the status for a work request

Before you begin



Note:

Work request status information can be used to schedule workflows and trigger events.

Procedure

1. In the navigator, click **Production**.
2. In the **Folders** list, select the resource type you want to work with.



Tip:

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. On the **Details** tab, from the **Status** drop-down list, select the current state of the work request.



Important:

Changing the status can alter another production process. Status changes are signaled in the **events model**.

5. Click **Save**.

Related reference

[Condition Events \(on page 524\)](#)

Work Request Parameters

The Requirements tab for the selected work request outlines all the information required to undertake a single production process. You can view and modify the parameter values for the current execution of the process.

You can also view and modify resource quantities for the material, equipment, and personnel resources in this process in a single location. Some resources are part of a larger, more generalized group or class. For these resources, you can select a more defined resource than the one that is currently defined as a requirement. For example, if the selected process indicates that it requires a mixer to perform a task, but the segment in this work request actually requires a *specific* mixer to complete the task at a particular time that this mixer is available, then you can indicate the specific mixer at this level.

The following table describes the attributes specific to this resource type.

Attribute	Value	Description
Expand All	N/A	Click to expand all the segment information for all the work requests' work definitions.
Collapse All	N/A	Click to collapse the associations.
Earliest Start Time	System-defined	Select the start date and time for the segment.
Latest End Time	System-defined	Select the end date and time for the segment.
Work Parameters: Value field	User-defined	Specifies a parameter for the segment.
All requirements: Quantity field	User-defined	Specifies a parameter for the segment.
Select button	User-defined	Click to open the Universal Browser to select a new resource requirement.



Note:
This button is available only if a class resource

Attribute	Value	Description
		 needs to be further defined.
Clear button	User-defined	Click to remove the selected resource requirement.
Material Requirements: Consumed	User-defined	Displays the output of the material resources when this work definition segment runs.
Material Requirements: Produced	User-defined	Displays material required for the operation of the work request.
Material Requirements: Consumable	User-defined	Displays information about the material resources that are consumed during the operation of the work request.

Configure time span for a segment

Before you begin

Procedure

1. In the navigator, click **Production**.
2. In the **Folders** list, select the resource type you want to work with.



Tip:

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. On the **Requirements** tab, select the segment you want to work with.
The segment expands displaying the requirement information.
5. In the **Earliest Start Time** and **Latest End Time** fields, select a date and time, and then click **OK**.
6. Click **Save**.

Modify parameter values for a segment

Before you begin

Procedure

1. In the navigator, click **Production**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. On the **Requirements** tab, select the segment you want to work with.
The segment expands displaying the requirement information.
5. In the **Value** field of the parameter you want to modify, enter a new value.
6. Click **Save**.

Modify resource quantities for a segment

Before you begin

Procedure

1. In the navigator, click **Production**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. On the **Requirements** tab, select the segment you want to work with.
The segment expands displaying the requirement information.
5. In the **Quantity** field of the resource requirement you want to modify, enter a new quantity.
6. If applicable, click **Select** to select a new resource from the Universal Browser.

**Tip:**

Click **Clear** to remove the resource.

7. Click **Save**.

Post-execution work request results

The Response Information tab contains the results of the selected, executed work request.

A work request can have more than one work response associated with it, and a work response can have zero or more segment responses associated with it. A segment response displays detailed information related to the execution of a work request, such as work data, time, location, and equipment, material, and personnel actual information.

**Tip:**

You can view these segment responses when they have been configured using the Call Method activity.

The following table describes the attributes specific to this resource type.

Attribute	Value	Description
Item List	User-defined	Displays a list of associated work responses that can be sorted by work response, segment response, work process segment, or location.
Query Segment Response	N/A	Click to display the results for an executed work request.

Query segment responses

Before you begin

**Important:**

To use this feature, response data for the work request must exist in the runtime database. You can create call method activities to view the segment responses.

Procedure

1. In the navigator, click **Production**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. On the **Response Information** tab, click **Query Segment Responses**.
All relevant work responses display in the Item List.

**Tip:**

You can sort this list by Work Response, Segment Response, Work Process Segment, or Location.

5. To view item specific information, select an item.
The information about the work response displays in the segment responses panel.

Work Requests View

The Work Requests View provides information at a glance of certain properties associated with production and equipment resources. You can also filter specific information related to these resources.

The following table describes the properties available in the Work Requests View. These properties are in a table that you can sort by clicking on a column heading. You can also update the table by clicking the Refresh button.

Property	Value	Description
Name	N/A	The name given to a resource when it was created. To view and edit this resource, select the name. The resource opens in the S95 Model Editor.
Description	System-defined	Specifies a display name for the selected property.

Property	Value	Description
Status	System-defined	Specifies the state of the work request and resources associated with the work request.
Work Type	System-defined	Specifies the category of work that the work definition is associated with.
Location	System-defined	Specifies the equipment resource associated with the work request.
Priority	System-defined	Specifies the order in which the work request is performed.
Work Definition	System-defined	Specifies the work definition that is associated with the work request.
Created	System-defined	Specifies the date and time that the work request was created.
Scheduled Start	System-defined	Specifies the date and time that the work request starts running.
Scheduled End	System-defined	Specifies the date and time that the work request stops running.

Filters

You can filter the items displayed in the main work request view by using the filtering feature. The following table describes the properties available that you can use to filter your work requests:

Property	Value	To set your filtering options...
Status	N/A	Select one or more check boxes.
Time Filters	System-defined	Select the applicable dates and times:

Property	Value	To set your filtering options...
		<ul style="list-style-type: none"> • Created after/Created before • Scheduled to start after/Scheduled to start before • Scheduled to end after/Scheduled to end before
Clear button	User-defined	Click to remove a filter.
Priority	System-defined	<p>Select one of the following options and, if applicable, enter the required information in the associated field(s):</p> <ul style="list-style-type: none"> • All • Exactly • Range (greater than or less than)

Related reference

[Work Data Classes \(on page 457\)](#)

Display work requests

Before you begin**Procedure**

1. Access the resource you want to work with:
 - a. From the navigator, select the model that contains the resource you want to work with.
In the next pane, a list of resources appears.
 - b. From this list, select the resource you want to work with.
 - c. If applicable, from the next list in the hierarchy, select the resource you want to work with.

**Note:**

As more lists appear, select the appropriate resource until you find the one you are looking for.

In the following model...	Click...
Production	Work Definitions , and then select the resource that has the work request you want to view.
Production	Work Requests .
Equipment	The resource associated with the work request you want to view.

2. In the **Show Compatible** panel, click **Work Requests View**.
3. To sort items, click a column heading.
4. To see newly created work requests, click **Refresh**.

Filter work requests

Before you begin

Procedure

1. Access the resource you want to work with:
 - a. From the navigator, select the model that contains the resource you want to work with. In the next pane, a list of resources appears.
 - b. From this list, select the resource you want to work with.
 - c. If applicable, from the next list in the hierarchy, select the resource you want to work with.

**Note:**

As more lists appear, select the appropriate resource until you find the one you are looking for.

In the following model...	Click...
Production	Work Definitions , and then select the resource that has the work request you want to view.
Production	Work Requests .
Equipment	The resource associated with the work request you want to view.

2. In the **Show Compatible** panel, click **Work Requests View**.
3. Click the **Show Filters** arrow.

**Tip:**

To return to the table view, click **Hide Filters**.

4. Set filters by using the following methods:
 - In the **Status** section, select one or more check boxes.
 - In the **Time Filters** section, select one or more dates and times from the drop-down arrows.
 - In the **Priority** section, select an option, and if applicable, enter values in the appropriate fields.
5. Click **Refresh** to activate the filters and return to the table view.

Events Model

Events Model

Events are a fundamental part of production. The events model is used to map out the critical events that occur to control production. These events can be created within this application or from external systems, and can be used to control workflow execution.

The event model receives external events, detects system events, and notifies applications that subscribe to events when an event has occurred. Within the event model, you can configure, monitor, and view reports on events. This component exposes a library of event definitions, as well as the means to define them. Events initiate work requests associated with specific events when they occur.

Event definitions can be user-defined or provided from another application or service provider. Event definitions are stored in the Directory of Resources and are available to subscribers, who are notified when an event has occurred.

There are two types of configurable events: condition events and time events. Conditional and time-based events are categorized and associated with elements from other models. The Events model also provides relevant event-based information on production and workflow events. You must enable an event to activate it. When it is enabled, it is monitored by the system until you disable or delete it.

Related reference

[Evaluation of Condition Expressions \(on page 525\)](#)

[Event Sources \(on page 532\)](#)

[Condition Event Expression Variables \(on page 529\)](#)

[Condition Expressions \(on page 531\)](#)

Enable an event

Procedure

1. In the navigator, click **Events**.
2. In the Folders list, select the resource type you want to work with.

**Tip:**

As more lists appears, select the appropriate resource until you find the one you want to work with.

3. Click **Enable**.

**Note:**

After the event is enabled, it is monitored by the system until it is disabled or deleted. If you make changes to an event, ensure that you save it before you enable the event; otherwise, the old event is enabled.

Disable an event

Procedure

1. In the navigator, click **Events**.
2. In the Folders list, select the resource type you want to work with.

**Tip:**

As more lists appears, select the appropriate resource until you find the one you want to work with.

3. Click **Disable**.**Note:**

When you disable an event, it is no longer triggered or monitored by the system.

Time Events

Time events can be set by time span or by calendar, and they can be either single or recurring events.

Events always fire against the local server time. However, the following exception applies during daylight saving time (DST) change.

Example: When time interval is configured in Days (or) in combination of Days and Hours.

- When the clock is set back in autumn, an event set for 2:30 A.M. fires only once when the time is reached for the second time (after DST ends).
- When the clock is moved forward in spring, an event set for 2:30 A.M. fires once at 3:30 A.M. on the first day (which would have been 2:30 A.M. before the time change), and then fired again at 2:30 A.M. the following day.

Example: When time interval is configured in Hours.

- When the clock is set back in autumn, an event set for a particular time, such as 1:30 A.M. fires twice - once at the original time of 1:30 A.M., and then once again when that time reaches 1:30 A.M. for the second time.
- **When time interval is configured as Hour(s):** When the clock is moved forward in spring, an event set for 2:30 A.M. fires at 3:30 A.M. on the first day (which would have been 2:30 A.M. before the time change), and then at 2:30 A.M. the following day.

**Note:**

When you change the timezone in your system, make sure to restart the Proficy Workflow Client to reflect the correct event time in the **Event Editor**.

Resource Properties

The following table describes general information on properties for user-defined resources:

Attribute	Value	Description
Name	Required; user-defined up to 50 characters	Specifies a display name for a resource.
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

Configure a time span event

Procedure

1. In the navigator, click **Events**.
2. In the Folders list, select the resource type you want to work with.



Tip:

As more lists appears, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click **Time Span Event**.
5. Click the arrows and select the day and time to set:
 - the time span
 - the start offset
6. Click **Save**.

Configure a calendar event

Procedure

1. In the navigator, click **Events**.
2. In the Folders list, select the resource type you want to work with.

**Tip:**

As more lists appears, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click **Calendar Event**.
5. In the **First Occurrence Date** section, click the arrows and select the month, day, and year.
6. In the **Event Time** section, click the arrows and select the hour, minutes, and seconds.
7. Click **Recurs until**, and then select the month, day, and year.
8. Optionally:

Click...	If the event occurs...
Recurs indefinitely	continuously
Does not recur	one time

9. To set a recurrence:

Type	Enter...
Daily	the number days between each occurrence.
Weekly	the number weeks between each occurrence, and then select the day(s) of the week of each occurrence.
Monthly	the exact day of the occurrence or select a repeating schedule, and then enter the number of months between each occurrence.

10. Click **Save**.

Time Span Event Attributes

Time span events allow you to indicate when you want an event to occur after the event has been enabled.

You indicate the initial start offset from midnight, as well as the time lapse between intervals. After this time span is set, it runs at this interval indefinitely. Events always fire against the local server time.

Example

If you set the start offset at five minutes and the time span at 60 minutes, then the time span event is every hour at five minutes after.

The following table describes time span events.

Attribute	Value	Description
Specify the time span between each event	System-defined	Indicates the length of time, in days, hours, minutes, or seconds, between each event.
Specify the start offset	System-defined	Indicates the length of time set to elapse from midnight, in days, hours, minutes, or seconds, before the first interval begins.

Calendar Event Attributes

Calendar events allow you to indicate a recurrence pattern. You can set a start date and time and an end date and time. Events always fire against the local server time.

Example

The first occurrence date does not have to match the recurrence setting. For example, if you set the First Occurrence Date to June 15, 2009 with a monthly recurrence set to the first Monday of every month, the event will trigger on June 15, 2009 even though it is not the first Monday of the month. The recurrence schedule determines the subsequent occurrences.

Event Range and Time

The following table describes ranges and times for calendar events.



Tip:

These settings work in conjunction with Recurrence Settings, if required.

Attribute	Value	Description
First Occurrence Date	System-defined	Indicates the month, day, and year on which an event first occurs.

Attribute	Value	Description
Event Time	System-defined	Indicates the hour, minute, and second at which an event first occurs.
Recur until	System-defined	Indicates the day, month, year that the event will end.
Recur indefinitely	System-defined	An event occurs continuously at the set date and time.
Does not recur	System-defined	An event occurs only once at the set date and time.

Recurrence Settings

These settings work in conjunction with event ranges and times. The following table describes recurrence settings for calendar events.

Attribute	Value	Description
Daily Recurrence	System-defined	An event occurs daily at a specific date and time unless there is an end date set.
Weekly Recurrence	System-defined	An event occurs weekly at a specific day and time unless there is an end date set. Select the day(s) of the week on which you want the event to occur and how many weeks between each occurrence.
Monthly Recurrence	System-defined	An event occurs monthly at a specific date and time unless there is an end date set. Select the frequency of the recurrence. You can specify the exact day within a month or months or select a repeating schedule such as the First Monday or every month.

Condition Events

Condition events are defined by logical expressions against variable values. When these components are set and enabled, they are monitored by the system. When expressions evaluate to true, the event occurs and subscribers to the event are notified.

Related reference

[Condition Expressions \(on page 531\)](#)

Properties and Attributes of Condition Events

The Details tab offers general information about the selected condition event.

You can change the name and description of this resource.

Attribute	Value	Description
Name	Required; user-defined up to 50 characters	Specifies a display name for a resource.
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

The following table describes the attributes specific to this resource type.

Attribute	Value	Description
State	User-defined	Indicates if the event is enabled or disabled.
Event Summary	N/A	Summarizes the current status of a conditional event, as well as when it will be used.
Variables Summary	N/A	Displays all variables associated with the selected event, if any. Variables can be bound to properties of other production resources.

Attribute	Value	Description
Condition Expressions Summary	N/A	Displays all expressions associated with the selected event, if any.

Evaluation of Condition Expressions

The Trigger Settings tab allows you to determine when the configured condition expressions will be evaluated. The results of these evaluations determine whether or not the event can run.

These settings are outlined when a **Create Event activity** in a workflow is created.

The following table provides more information on the outcome of selecting a trigger type:

Trigger	Value	Description
Trigger only on Variable Value changes	System-defined	Evaluates the expressions when a variable changes. These changes can occur through a bind to external data or to an attribute of another event.
Trigger on Event or Variable Value changes	System-defined	Evaluates the expressions when a variable changes or when the selected event trigger is signaled.
Trigger only on Event	System-defined	Evaluates the expressions only when the selected event trigger is signaled.
Set Event Trigger	N/A	Click to open the Configure Event Trigger dialog box and select the appropriate event trigger.
Clear	N/A	Click to clear the selected event trigger.

Set a trigger for a conditional event

Procedure

1. In the navigator, click **Events**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

- In the **Displays** panel, select an editor.

The editor appears in the workspace, displaying the resource information.

- On the **Trigger Settings** tab, select a trigger type.

Select...	Then...
Trigger only on Variable Value changes	go to step 6.
Trigger on Event or Variable Value changes	go to step 4.
Trigger only on Event	go to step 4.

- Click **Set Event Trigger**.

The Configure Event Trigger dialog box appears.

- Select the event trigger you want to use, and then click **OK**.
- Click **Save**.

Set up class-based events

You can bind class properties to event variables so that an event can detect a property change in any equipment to which that class has been added.

About this task

**Note:**

This does not allow you to detect property changes in equipment properties where that class has been used as the Type.

Procedure

- In the navigator, click **Events**.
- In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. On the **Trigger Settings** tab, select either:
 - **Trigger only on Variable Value changes**
 - **Trigger on Event or Variable Value changes**
5. Click the **Variables** tab, and then click **Add**.
6. Configure the variable as follows:
 - a. In the **Variable Name** field, enter a name.

**Note:**

The variable name must be alphanumeric and must not contain spaces.

- b. Select **Bind variable to external data**, and then click **Configure Data Item**.
The **Configure Variable Data Source** dialog box appears.
- c. Select an equipment class, and then click **Next**.

**Note:**

A class property is editable and can be deleted even when it is being monitored by an enabled condition event; however, the event is invalid if the class property is changed or deleted from the equipment class.

- d. In the Item specific configuration section, click the ellipsis button. The **Data Source Browser** appears.
- e. Select an equipment class property, and then click **OK**.
- f. Click **Finish**, and then click **Save**.

Create variables for a condition event

Procedure

1. In the navigator, click **Events**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.

The editor appears in the workspace, displaying the resource information.

4. On the **Variables** tab, click **Add**.



Tip:

To permanently remove a variable from the event, select the variable, and then click **Delete**.

5. In the **Variable Name** field, enter a name.



Note:

The variable name must be alphanumeric and must not contain spaces.

6. Select one of the following options:

Select...	Then...
Bind variable to external data	<ul style="list-style-type: none"> ◦ Click Configure Data Item. The Configure Variable Data Source dialog box appears. ◦ Configure the data source. ◦ Click OK.
Bind variable to an attribute of the Event Trigger	<div data-bbox="873 1108 920 1157" data-label="Image"> </div> <p>Note: This option is available only if a trigger event has been set (on page 525).</p> <ul style="list-style-type: none"> ◦ Select an attribute. ◦ Enter a default value.
Variable represents a constant value	<ul style="list-style-type: none"> ◦ Select a data type. ◦ Enter a value for the data type.

7. Click **Save**.

Create an expression for a condition event

Before you begin

For more information on how to build expressions, see the **expressions editor**.

**Note:**

Each expression must evaluate to True or False (Boolean).

Procedure

1. In the navigator, click **Events**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. On the **Condition Expressions** tab, click **Add**.

**Tip:**

To permanently remove an expression from the event, select the expression, and then click **Delete**.

5. To modify the name and description, click **Edit**. In the **Name** and **Description** fields, enter a name and description.
6. Double-click expression functions, expression variables, and/or expression operators to add them to the **Enter Expression** field, and then enter user-defined values.
7. Select the **Latch Expression** check box to latch the expression.
8. Click **Check Syntax** to check the expression and condition for configuration errors.
9. Click **Test** to evaluate the expression.
The Test Expression dialog box appears.
10. Click **Evaluate**.
11. Click **Close**, and then click **Save**.

Condition Event Expression Variables

The Variables tab allows you to add, modify, and delete expression variables to use when building event expressions. You can define one or more variables, which can be a constant value bound to an attribute of the trigger event, or bound to external data.

Variables are the input data used to build expressions. They can be literal values, represent an attribute of the event trigger, or can be bound to a data source.

Attribute	Value	Description
List	User-defined	Displays a list of variables that can be arranged by name.
Add	N/A	Click to add a new variable for use in expressions.
Delete	N/A	Click to delete the selected variable.
Name	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 40 characters that cannot contain spaces	Specifies the name for a variable.
Bind variable to external data	User-defined alphanumeric string (A-Z, a-z, 0-9)	Select this option to create a variable that is bound to an external data source.
Configure Data Item	N/A	<p>Select an external data source from those displayed on the Navigate tab, and specify additional input and output values for the data item's read and write functions.</p> <div data-bbox="1024 1199 1419 1509" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: Optional configuration information for the selected data item will vary, depending on the item that you select.</p> </div>
Bind variable to an attribute of the event trigger	System-defined	Select this option to create a variable that is bound to an attribute of the event trigger.
Attribute	User-defined	Select the trigger event attribute to which the variable is bound.
Data Type	System-defined	Displays appropriate settings based on the attribute chosen.

Attribute	Value	Description
Default Value	System-defined	Specifies the input value for the attribute.
Variable represents a constant value	System-defined	Select this option to create a variable that represents a constant value.
Data Type	User-defined	Select the value of the data type. <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: For more information about the data types and their values, see Data Types. </div>
Value	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 50 characters	Specifies a value for the data type selected.

Condition Expressions

The Condition Expressions tab allows you to create expressions using mathematical or Boolean logic.

When an event is triggered, the expressions are evaluated. The event's variables determine if an expression is true or false. The results of this evaluation must be all or partially true, depending on the event setting, in order for the event to occur.



Tip:

For general information on building expressions, see the **expression editor**.

The following table displays the attributes that are available in the events expression editor:

Attribute	Value	Description
Latch Expression	System-defined	Select this option if you want an expression to remain true after it has been evaluated, but is waiting for all other expression to be-

Attribute	Value	Description
		come true. When all expressions are true, the event is signaled.
Edit	User-defined	Displays the name and description of the selected expression.
Check Syntax	System-defined	Click to confirm if an expression has been completed and ordered correctly.
Test	System-defined	Click to evaluate the expression. The Test Expression dialog box appears.
Expression Functions tab	N/A	Displays the functions available to build your expression. Double-click a function to insert into your expression.
Expression Variables tab	N/A	Displays the variables available to build your expression. Double-click a variable to insert into your expression.

Related reference

[Condition Events \(on page 524\)](#)

Related information

[Events Model \(on page 517\)](#)

Event Sources

Event sources are data items that you can subscribe to.

When you configure the scheduled settings in a **workflow schedule** or the trigger settings of a **condition event**, event source properties are available to define further.

**Important:**

Available event sources depend on the service providers you have installed.

Event Source	Value	Description
Equipment	User-defined	If no equipment resources with property values have been added in the Equipment model, this event source is not available.
User-defined events	User-defined	If no time or condition event resources have been added in the Event model, this event source is not available.
Production	System-defined	<p>A production event is signaled when one of the following occurs:</p> <ul style="list-style-type: none"> • New Work Request • Work Request Status Changed
Workflow	System-defined	<p>A workflow event is signaled when one of the following occurs:</p> <ul style="list-style-type: none"> • Task Expired • Task Step Expired
GE Historian	User-defined server items	<p>If no server items have been added in the Proficy System model, this event source is not available.</p> <p>A GE Historian event is signaled when the following occurs:</p> <ul style="list-style-type: none"> • Tag Synchronization Event
Material	System-defined	A material event is signaled when one of the following occurs:

Event Source	Value	Description
		<ul style="list-style-type: none"> • Material Lot Deleted • Material Lot HasChildren Changed • Material Lot Location Changed • Material Lot Quantity Changed • Material Lot Status Changed • Material Sublot Deleted • Material Sublot HasChildren Changed • Material Sublot Location Changed • Material Sublot Quantity Changed • Material Sublot Status Changed • New Material Lot Created • New Material Sublot Created
Compliance	System-defined	<p>A compliance event is signaled when the following occurs:</p> <ul style="list-style-type: none"> • Electronic Signature Signing Result
Security	System-defined	<p>A security event is signaled when one of the following occurs:</p> <ul style="list-style-type: none"> • SSO Authorization Failed (executes when a single sign on user is not configured correctly)

Event Source	Value	Description
		<ul style="list-style-type: none"> The account is locked out (executes when a user account is locked out) User Account Created (executes when a new user account is created)

User-Defined Event Source Properties

You can use event source properties to monitor particular events related to conditional events or workflow schedules.



Note:

You can only bind parameters to property values of similar data types.

Common Event Source Properties

The following table describes the common properties available for each event source.

Property	Data Type	Description
Id	GUID	Unique key that identifies the event definition in the data model.
Name	String	Display name of the event definition resource.
Description	String	Additional information related to the event definition resource.
Timestamp	DateTime	The actual UTC time that the event occurred.
EventNodeAddress	EventAddress	The Directory Address of the event that occurred.
EventType	String	The .NET type of the event occurrence.

Property	Data Type	Description
KeepArchive	Boolean	If the event associated with the workflow schedule is maintained in history by the event manager, this is True.

Condition Event Properties

Following are all available properties that are specific to this event source.

Property	Value	Description
Last Changed Variable	String	Specifies the name of the last variable that changed in the condition event.
Other	User-defined	Specifies a collection of all the variables of an event. These variables are dependent on the resources that a user has previously created. Variables associated with condition events become properties that you can use to bind to other data items.

Time Event Properties

Following are all available properties that are specific to this event source.

Property	Value	Description
Other	User-defined	Specifies a collection of all the variables of an event. These variables are dependent on the resources that a user has previously created.

Production Event Source Properties

You can use event source properties to monitor particular events related to conditional events or workflow schedules.



Note:

You can only bind parameters to property values of similar data types.

Common Event Source Properties

The following table describes the common properties available for each event source.

Property	Data Type	Description
Id	GUID	Unique key that identifies the event definition in the data model.
Name	String	Display name of the event definition resource.
Description	String	Additional information related to the event definition resource.
Timestamp	DateTime	The actual UTC time that the event occurred.
EventNodeAddress	EventAddress	The Directory Address of the event that occurred.
EventType	String	The .NET type of the event occurrence.
KeepArchive	Boolean	If the event associated with the workflow schedule is maintained in history by the event manager, this is True.

Event Source Work Request Properties

New Work Request

This event is signalled when a work request is created.

Following are all available properties that are specific to this event source.

Property	Data Type	Description
WorkDefinition	DirectoryResource	Specifies the work definition associated with the work request that was created or changed (depending on the event). It can be used to determine the appropriate workflow to execute in order to process the work request.
WorkRequest	DirectoryResource	Specifies the work request that was created, which is related to the event associated with the workflow schedule.
WorkRequestS95ID	String	Specifies the unique identifier of the work request.
WorkType	String	Specifies the category of work, such as Production, Maintenance, Quality, Inventory, and Mixed.
Attributes	WorkRequestAttributes	Contains the values of the attributes of the work request.

Work Request Status Changed

This event is signaled when the status of an existing work request changes.

Following are all available properties that are specific to this event source.

Property	Data Type	Description
WorkRequest	DirectoryResource	Specifies the work request that changed status, which is related to the event associated with the workflow schedule.
WorkDefinition	DirectoryResource	Specifies the work definition associated with the work request

Property	Data Type	Description
		that was created or changed (depending on the event). It can be used to determine the appropriate workflow to execute in order to process the work request.
WorkRequestS95ID	String	Specifies the unique identifier of the work request.
PreviousStatus	String	Specifies the status of the work request before the change.
NewStatus	String	<p>Specifies the updated status of the work request. The possible statuses include:</p> <ul style="list-style-type: none"> • Draft • Unscheduled • Pending • Running • Paused • Completed • Failed • Canceled • Unknown
WorkType	String	Specifies the category of work, such as Production, Maintenance, Quality, Inventory, and Mixed.
Attributes	WorkRequestAttributes	Contains the values of the attributes of the work request.
Location	DirectoryResource	Specifies the element in the equipment hierarchy that is associated with the work request.

GE Historian Event Source Properties

You can use event source properties to monitor particular events related to conditional events or workflow schedules.



Note:

You can only bind parameters to property values of similar data types.

Common Event Source Properties

The following table describes the common properties available for each event source.

Property	Data Type	Description
Id	GUID	Unique key that identifies the event definition in the data model.
Name	String	Display name of the event definition resource.
Description	String	Additional information related to the event definition resource.
Timestamp	DateTime	The actual UTC time that the event occurred.
EventNodeAddress	EventAddress	The Directory Address of the event that occurred.
EventType	String	The .NET type of the event occurrence.
KeepArchive	Boolean	If the event associated with the workflow schedule is maintained in history by the event manager, this is True.

Event Source Tag Synchronization Event

This event is signalled when the current tag status and event details on the selected Historian server changes. When tags are added or applied, this event occurs.

Following are all available properties that are specific to this event source.

Property	Data Type	Description
Status	String	Specifies the current status of the synchronization process.
Message	String	A user message describing what occurred during the synchronization process.
ServerAddress	String	Specifies the server that changed.
TagName	String	Specifies the tag that was added.
CollectorName	String	Specifies the collector owner of the tag.

Workflow Event Source Properties

Event sources are data items that you can subscribe to when you configure a workflow schedule. You use event sources to define when the selected workflow schedule is triggered.

When an event source is added to a workflow schedule, you can bind the parameters from the attached workflow to event properties.



Note:

You can only bind parameters to property values of similar data types.

Common Event Source Properties

The following table describes the common properties available for each event source.

Property	Data Type	Description
Id	GUID	Unique key that identifies the event definition in the data model.
Name	String	Display name of the event definition resource.
Description	String	Additional information related to the event definition resource.

Property	Data Type	Description
Timestamp	DateTime	The actual UTC time that the event occurred.
EventNodeAddress	EventAddress	The Directory Address of the event that occurred.
EventType	String	The .NET type of the event occurrence.
KeepArchive	Boolean	If the event associated with the workflow schedule is maintained in history by the event manager, this is True.

Equipment Event Source Properties

You can use event source properties to monitor particular events related to conditional events or workflow schedules.

Common Event Source Properties

The following table describes the common properties available for each event source.

Property	Data Type	Description
Id	GUID	Unique key that identifies the event definition in the data model.
Name	String	Display name of the event definition resource.
Description	String	Additional information related to the event definition resource.
Timestamp	DateTime	The actual UTC time that the event occurred.
EventNodeAddress	EventAddress	The Directory Address of the event that occurred.

Property	Data Type	Description
EventType	String	The .NET type of the event occurrence.
KeepArchive	Boolean	If the event associated with the workflow schedule is maintained in history by the event manager, this is True.

Event Source Class Added to Instance

This event occurs when a class resource type is added to a piece of equipment.

Following are all available properties that are specific to this event source.

Property	Data Type	Description
EquipmentAddress	DirectoryResource	Specifies the address of the equipment that changed.
ChangeType	String	"Class Added" string that identifies this event.
Change	String	Specifies the name of the class that was added.
ChangeAddress	DirectoryResource	Specifies the address of the class that was added.

Event Source Class Removed from Instance

This event occurs when a class resource type is removed from a piece of equipment.

Following are all available properties that are specific to this event source.

Property	Data Type	Description
EquipmentAddress	DirectoryResource	Specifies the address of the equipment that changed.
ChangeType	String	"Class Removed" string that identifies this event.

Property	Data Type	Description
Change	String	Specifies the name of the class that was removed.
ChangeAddress	DirectoryResource	Specifies the address of the class that was removed.

Material Event Source Properties

You can use event source properties to monitor particular events related to conditional events or workflow schedules.



Note:

You can only bind parameters to property values of similar data types.

Common Event Source Properties

The following table describes the common properties available for each event source.

Property	Data Type	Description
Id	GUID	Unique key that identifies the event definition in the data model.
Name	String	Display name of the event definition resource.
Description	String	Additional information related to the event definition resource.
Timestamp	DateTime	The actual UTC time that the event occurred.
EventNodeAddress	EventAddress	The Directory Address of the event that occurred.
EventType	String	The .NET type of the event occurrence.
KeepArchive	Boolean	If the event associated with the workflow schedule is maintained

Property	Data Type	Description
		in history by the event manager, this is True.

Event Source Material Lot Deleted

Signalled when a lot is deleted, not when a lot is removed by archive and purge.

Following are all available properties that are specific to this event source.

Property	Data Type	Description
S95 ID	GUID	Unique key that identifies the lot deleted.
Material Definition	DirectoryResource	Specifies the material definition of the lot.
Location	DirectoryResource	Specifies the equipment instance associated with the lot.
Status	String	Specifies the state of the lot.
Quantity	Double	Specifies the quantity of the lot.
UoM (Unit of Measure)	String	Specifies the unit of measure for the lot quantity.

Event Source Material Lot Has Children Changed

Signalled when the first child of a lot is created or when the last child of a lot is removed.

Following are all available properties that are specific to this event source.

Property	Data Type	Description
S95 ID	GUID	Unique key that identifies the lot where the first child was created or the last child was removed.
Material Definition	DirectoryResource	Specifies the material definition of the lot.
Location	DirectoryResource	Specifies the equipment instance associated with the lot.

Property	Data Type	Description
Status	String	Specifies the state of the lot.
Quantity	Double	Specifies the quantity of the lot.
UoM (Unit of Measure)	String	Specifies the unit of measure for the lot quantity.

Event Source Material Lot Location Changed

Signalled when the equipment instance association or location of a lot changes.

Following are all available properties that are specific to this event source.

Property	Data Type	Description
S95 ID	GUID	Unique key that identifies the lot where the first child was created or the last child was removed.
Material Definition	DirectoryResource	Specifies the material definition of the lot.
New Location	DirectoryResource	Specifies the equipment instance currently associated with the lot.
Old Location	DirectoryResource	Specifies the equipment instance previously associated with the lot.
Status	String	Specifies the state of the lot.
Quantity	Double	Specifies the quantity of the lot.
UoM (Unit of Measure)	String	Specifies the unit of measure for the lot quantity.

Event Source Material Lot Quantity Changed

Signalled when the quantity of a lot changes.

Following are all available properties that are specific to this event source.

Property	Data Type	Description
S95 ID	GUID	Unique key that identifies the lot where the first child was created or the last child was removed.
Material Definition	DirectoryResource	Specifies the material definition of the lot.
Location	DirectoryResource	Specifies the equipment instance associated with the lot.
Status	String	Specifies the state of the lot.
New Quantity	Double	Specifies the current quantity of the lot.
Old Quantity	Double	Specifies the previous quantity of the lot.
UoM (Unit of Measure)	String	Specifies the unit of measure for the lot quantity.

Event Source Material Lot Status Changed

Signalled when the status of a lot changes.

Following are all available properties that are specific to this event source.

Property	Data Type	Description
S95 ID	GUID	Unique key that identifies the lot where the first child was created or the last child was removed.
Material Definition	DirectoryResource	Specifies the material definition of the lot.
Location	DirectoryResource	Specifies the equipment instance associated with the lot.
New Status	String	Specifies the current state of the lot.

Property	Data Type	Description
Old Status	String	Specifies the previous state of the lot.
Quantity	Double	Specifies the quantity of the lot.
UoM (Unit of Measure)	String	Specifies the unit of measure for the lot quantity.

Event Source Material Sublot Deleted

Signaled when a subplot is deleted, not when a subplot is removed by archive and purge.

Following are all available properties that are specific to this event source.

Property	Data Type	Description
S95 ID	GUID	Unique key that identifies the subplot deleted.
Lot ID	GUID	Unique key that identifies the parent lot of the subplot.
Material Definition	DirectoryResource	Specifies the material definition of the lot.
Location	DirectoryResource	Specifies the equipment instance associated with the subplot.
Status	String	Specifies the state of the subplot.
Quantity	Double	Specifies the quantity of the subplot.
UoM (Unit of Measure)	String	Specifies the unit of measure for the subplot quantity.

Event Source Material Sublot Has Children Changed

Signalled when the first child of a subplot is created or when the last child of a subplot is removed.

Following are all available properties that are specific to this event source.

Property	Data Type	Description
S95 ID	GUID	Unique key that identifies the subplot where the first child was created or the last child was removed.
Lot ID	GUID	Unique key that identifies the parent lot of the subplot.
Material Definition	DirectoryResource	Specifies the material definition of the lot.
Location	DirectoryResource	Specifies the equipment instance associated with the subplot.
Status	String	Specifies the state of the subplot.
Quantity	Double	Specifies the quantity of the subplot.
UoM (Unit of Measure)	String	Specifies the unit of measure for the subplot quantity.

Event Source Material Sublot Location Changed

Signalled when the equipment location of a subplot changes.

Following are all available properties that are specific to this event source.

Property	Data Type	Description
S95 ID	GUID	Unique key that identifies the subplot where the first child was created or the last child was removed.
Lot ID	GUID	Unique key that identifies the parent lot of the subplot.
Material Definition	DirectoryResource	Specifies the material definition of the lot.

Property	Data Type	Description
New Location	DirectoryResource	Specifies the equipment instance currently associated with the subplot.
Old Location	Directory Resource	Specifies the equipment instance previously associated with the subplot.
Status	String	Specifies the state of the subplot.
Quantity	Double	Specifies the quantity of the subplot.
UoM (Unit of Measure)	String	Specifies the unit of measure for the subplot quantity.

Event Source Material Sublot Quantity Changed

Signalled when the quantity of a subplot changes.

Following are all available properties that are specific to this event source.

Property	Data Type	Description
S95 ID	GUID	Unique key that identifies the subplot where the first child was created or the last child was removed.
Lot ID	GUID	Unique key that identifies the parent lot of the subplot.
Material Definition	DirectoryResource	Specifies the material definition of the lot.
Location	DirectoryResource	Specifies the equipment instance associated with the subplot.
Status	String	Specifies the state of the subplot.
New Quantity	Double	Specifies the current quantity of the subplot.

Property	Data Type	Description
Old Quantity	Double	Specifies the previous quantity of the subplot.
UoM (Unit of Measure)	String	Specifies the unit of measure for the subplot quantity.

Event Source Material Sublot Status Changed

Signalled when the status of a subplot changes.

Following are all available properties that are specific to this event source.

Property	Data Type	Description
S95 ID	GUID	Unique key that identifies the subplot where the first child was created or the last child was removed.
Lot ID	GUID	Unique key that identifies the parent lot of the subplot.
Material Definition	DirectoryResource	Specifies the material definition of the lot.
Location	DirectoryResource	Specifies the equipment instance associated with the subplot.
New Status	String	Specifies the current state of the subplot.
Old Status	String	Specifies the previous state of the subplot.
Quantity	Double	Specifies the quantity of the subplot.
UoM (Unit of Measure)	String	Specifies the unit of measure for the subplot quantity.

Event Source New Material Lot Created

Signaled when a new lot is created.

Following are all available properties that are specific to this event source.

Property	Data Type	Description
S95 ID	GUID	Unique key that identifies the lot created.
Material Definition	DirectoryResource	Specifies the material definition of the lot.
Location	DirectoryResource	Specifies the equipment instance associated with the lot.
Status	String	Specifies the state of the lot.
Quantity	Double	Specifies the quantity of the lot.
UoM (Unit of Measure)	String	Specifies the unit of measure for the lot quantity.

Event Source New Material Sublot Created

Signaled when a new subplot is created.

Following are all available properties that are specific to this event source.

Property	Data Type	Description
S95 ID	GUID	Unique key that identifies the subplot created.
Lot ID	GUID	Unique key that identifies the parent lot of the subplot.
Material Definition	DirectoryResource	Specifies the material definition of the lot.
Location	DirectoryResource	Specifies the equipment instance associated with the subplot.
Status	String	Specifies the state of the subplot.
Quantity	Double	Specifies the quantity of the subplot.

Property	Data Type	Description
UoM (Unit of Measure)	String	Specifies the unit of measure for the subplot quantity.

Compliance Event Source Properties

You can use event source properties to monitor particular events related to conditional events or workflow schedules.



Note:

You can only bind parameters to property values of similar data types.

Common Event Source Properties

The following table describes the common properties available for each event source.

Property	Data Type	Description
Id	GUID	Unique key that identifies the event definition in the data model.
Name	String	Display name of the event definition resource.
Description	String	Additional information related to the event definition resource.
Timestamp	DateTime	The actual UTC time that the event occurred.
EventNodeAddress	EventAddress	The Directory Address of the event that occurred.
EventType	String	The .NET type of the event occurrence.
KeepArchive	Boolean	If the event associated with the workflow schedule is maintained in history by the event manager, this is True.

Event Source Electronic Signature Signing Result

This event is signalled when an electronic signature signing succeeded, failed, or was cancelled. Each event source property is based on a single instance of an electronic signature.

Following are all available properties that are specific to this event source.

Property	Data Type	Description
SignatureID	ESignatureID	Specifies a value that identifies a particular signature.
SigningDescription	String	Specifies the displayed description of the electronic signature at the time of signing.
RequiresVerifier	Boolean	Specifies if the signature requires that a verifier and performer sign for the signature to complete successfully.
IsSuccessful	Boolean	Specifies if the signing completed successfully.
IsCancelled	Boolean	Specifies if the signature was cancelled by a user (by clicking the Cancel Signature button).
IsFailed	Boolean	Specifies if the signature failed because the role signing attempt limit was reached.
ValidPerformersGroup	DirectoryResource	Specifies the directory resource for the security group that a valid performer must be part of to complete the signing successfully.
ValidVerifiersGroup	DirectoryResource	Specifies the directory resource for the security group that a valid verifier must be part of to complete the signing successfully.

Property	Data Type	Description
PerformerUsername	String	Specifies the login name of the person that signed as the performer of this signature.
Performer	DirectoryResource	Specifies the directory resource of the person that signed as the performer of this signature.
IsPerformerValid	Boolean	Specifies if a valid user signed as the performer.
VerifierUsername	String	Specifies the login name of the person that signed as the verifier for this signature.
Verifier	DirectoryResource	Specifies the directory resource of the person that signed as the verifier of this signature.
IsVerifierValid	Boolean	Specifies if a valid user signed as the verifier (only if a verifier was required for this signing).

Event View

Event View

The Event View displays multiple active condition and time events in the Proficy System and the resources that currently are attached to the events.

In this view, you can temporarily subscribe to an event to obtain live event information.

Overview Tab

The following table describes the available options.

Attribute	Value	Description
Event List	User-defined	Displays a list of the active condition and time events, and indicates the quantity of

Attribute	Value	Description
		each. Events can be sorted and grouped by name, event type, or logging enabled status.
Refresh	N/A	Click to retrieve the current set of events.
Subscribers	User-defined	Displays a list of resources that subscribe to the selected event.
Start Logging/Stop Logging	N/A	Click to start and stop logging event information.

Event Log Tab

The Event Log tab displays all the currently enabled events that have been triggered. This log lists up to 200 events; the oldest events drop off the list as new ones are added.

The following table describes the available options.

Attribute	Value	Description
Clear Log	N/A	Click to clear the event log.
Name	User-defined	Displays the name given to the event when it was created.
Timestamp	N/A	Displays the local time that the event was triggered.
Property List	N/A	Displays all properties, such as variables, of the event.

Start and stop logging event information

Before you begin



Note:

You must have created time or condition events and configured other resources to subscribe to them in order to view them in the Event List.

Procedure

1. In the navigator, click **Proficy System > Proficy System**.
2. In the **Displays** panel, click the working area you require.
3. From the **Event List** area, select the event you want to log.
The **Subscribers** area displays the resources that subscribe to the event.
4. From the **Subscribers** list, click **Start Logging**.
Event information for the selected event displays in the **Event Log** area.
5. To stop logging event information, click **Stop Logging**.
6. To clear the Event Log area, click **Clear Log**.

Actuals

Actuals

Resource actuals show the as-performed actual values for a resource during a production cycle. A resource actual is associated with a resource specification, making it easy to compare actual and specified values for properties.

When the planned use of resources does not match the actual use of resources, then an actual is used. The actual records the true values of resources during production, which can be compared to expected use. Work process segments have specifications, but no way to record actual values. Using actuals makes it possible to take a true measurement of the resources that were used.

Each actual corresponds to an existing equipment, material, or personnel resource.

IProductionRuntime Service Interface

Creating actuals and editing their properties is managed using call methods of the IProductionRuntime Service Interface. Form more information about these properties, see **Production Runtime Call Methods**.

Example

Suppose you have a workflow for attaching a wheel to a hub. You specify in the work process definition that during the production process, one wheel/rim assembly and six lug nuts are consumed. Now, you want to write an application that tracks your inventory of lug nuts, so you can replace them when needed. One way to do that would be to simply subtract six lug nuts from inventory every time the process runs. But sometimes a lug nut is defective, or was stripped while the operator tried to add it, and another one is needed. Now seven lug nuts have been consumed. With an actual, you can record that seven lug nuts were used instead of six, so the inventory count can be maintained accurately.

By having these values associated with the specifications, you can also tell when your production is in control. If your inventory seems to be shrinking faster than it should, you can write queries to return the difference between actual and specified values for the last month's work requests. You can use the queries to spot the point where you are using more resources than you specified. You can then drill down through the work responses looking for correlations with other specifications. Maybe you find that the spoilage rate for lug nuts goes up whenever an operator used air hammer #2. So you order an inspection of the hammer, and find out that it's not sealing the lug nuts correctly due to a flaw, and fix it.

Specification properties and their actual properties go a level deeper. Suppose you specify that a beer batch must cook for 2 hours at 135 degrees. Then your final beer batch goes through quality assurance and does not pass inspection. How can you determine what went wrong? You can run a query that compares actual measured values against specification values, to find anomalies. Your query shows that, the actual measured temperature during cooking was 110 degrees, not the specified 135 degrees. So you inspect the cooker, and find out it has a defective thermostat.

Material Actuals

Material actuals identify a material quantity used during a selected work process segment.

Resource actuals show the as-performed actual values for a resource during a production cycle. A resource actual is associated with a resource specification, making it easy to compare actual and specified values for properties.

When the planned use of material resources does not match the actual use of material resources, then a material actual is used. The material actual records the true values of resources during production, which can be compared to expected use. Work process segments have specifications, but no way to record actual values. Using material actuals makes it possible to take a true measurement of the resources that were used. For example, if the production normally used a specific batch of raw material and a separate batch was used, then a material actual would record the use of the other batch.

A material actual corresponds to an existing material definition resource.

IProductionRuntime Service Interface

Creating material actuals and editing their properties is managed using the **CreateMaterialActual_IProductionRuntime** and **AddResourceActualProperties_IProductionRuntime** call methods.

Property Groups and Aspects

Property Groups

Property groups are containers for properties. They allow you to read from or write to the properties of a resource.

A property group does not contain links to actual properties; it merely contains property names stored as strings.

You can use property groups with the equipment, material, personnel models, as well as OPC client and Historian servers.

Aspects

Aspects are a mechanism for dynamically extending existing resources by adding new attributes and business logic without modifying the original objects or recompiling them.

Aspects can be used to extend the attributes of an existing object, or they can link two or more disparate entities.

**Note:**

Creating aspects is for advanced users only. For assistance creating aspects, contact GE Digital.

Chapter 8. Data Administration

Data Administration Introduction

Email Server Configuration

You use the Email Editor to manage the email server configuration. From the email server, you can send out emails from a particular sender automatically.

1. In the navigator, click **Proficy System > Proficy System**.
2. From the **Displays** panel, select **HW/SW > Email** to display the **Email Configuration Setup** page.



Note:

The **Email Configuration Setup** page appears.

3. Enter information into the following fields, and then click **Save**.

Property	Value	Description
Server host name	User-defined	Specifies the host name or IP address of the SMTP server.
Enable SSL	N/A	Enables SSL encrypted communication to SMTP server.
Server port number	User-defined	Specifies the SMTP server's port number.
Anonymous	N/A	Select this option if you have anonymous access to the server. Selecting this option disables the User Name and Password entries.
User/Password	N/A	Select this option if you want to use user authentication when sending an email message.
User Name	User-defined	Account user name used to authenticate to SMTP server.

Property	Value	Description
Password	User-defined	Account password used to complete authentication to SMTP server.
Server timeout value	User-defined	Enter the number of seconds allowed to connect to the SMTP server before timing out.
Default sender address	User-defined	Enter the email address from which all emails will be sent.
Default sender name	User-defined	Optional: Enter a name related to the email account from which emails are being sent.
Test recipient email address	User-defined	Enter a recipient's email address in order verify the email server configuration.
Send Test Message	N/A	Click this button to test the email server configuration settings. If the configuration is correct, the recipient will receive a message from <i>Workflow Server</i> .

Proficy System

The Proficy System model is a container for elements that specify a system's physical configuration, specifically related to connectivity. The structure of the Proficy System model presents a metaphor for a real world computer and network topology.

The Proficy System model allows the system to connect to multiple data sources outside of the current configuration of Workflow. The Proficy System model also tracks clients connected to the server, other Workflow servers, and includes management of your current project, such as archive management and import/export of project data.

Available data sources consist of GE Historian, MS SQL, and OPC connections.

Electronic Signatures and Audit Trails

Electronic Signatures and Audit Trails

Workflow provides the ability to electronically sign forms in the run-time environment and to create an audit trail of those signatures, as well as audit other configuration changes made to the system.

You can use electronic signatures to create a more secure environment by requiring that operators electronically sign for data entry changes on forms in the Task List. An electronic signature uniquely identifies the operator making the change, and can optionally require the electronic signature of another person to verify the change. Operators no longer need to use paper and pen to record and sign for their actions, and the possibility of losing or damaging such records is essentially eliminated.

An audit trail is necessary for regulatory compliance, because it provides a method of recording actions that have been performed in the Workflow system. An audit trail helps to prove that you have control of your process and allows you to track not only what occurred, but when it occurred, and who was responsible for the occurrence. For example, an audit trail can be used to assist with an investigation of a product discrepancy.

Related reference

[21 CFR Part 11 Overview \(on page 566\)](#)

[Configure Audit Trails and Electronic Signatures \(on page 567\)](#)

Related information

[Audit Trails \(on page 567\)](#)

[Purpose of Electronic Signatures \(on page 562\)](#)

[Reporting \(on page 662\)](#)

[Reporting Database Replication \(on page 663\)](#)

Purpose of Electronic Signatures

Electronic signatures are intended for application developers, process control engineers, and workflow users who want or need to incorporate electronic signatures and audit trails into their operations.

Electronic signatures in Workflow show who performed a specific action within a certain process, and thereby legally identify the user who signed for an action.

This section assists users, operators, and supervisors responsible for creating a secure, auditable environment, especially those working with the 21 CFR Part 11 United States FDA government regulation.

Detailed permanent records of operator actions are written to, and stored in, your SQL Server database. You can query and report on these records, and then use this data to provide a comprehensive audit trail detailing the history of your process. For more information, see **Audit Trails**.

**Important:**

Using electronic signatures by itself does not ensure 21 CFR Part 11 compliance; however, electronic signatures can help provide the necessary electronic verification needed to satisfy the requirements of this regulation. For more information, see **21 CFR Part 11 Overview**.

Related reference

[How to Use Electronic Signatures \(on page 563\)](#)

[21 CFR Part 11 Overview \(on page 566\)](#)

[Signatures During Workflow Run Time \(on page 565\)](#)

[Electronic Signature Tracking \(on page 565\)](#)

[Configure Audit Trails and Electronic Signatures \(on page 567\)](#)

Related information

[Configure failure limits for electronic signatures \(on page 573\)](#)

[Electronic Signatures and Audit Trails \(on page 562\)](#)

[Audit Trails \(on page 567\)](#)

How to Use Electronic Signatures

Electronic signatures are the computer-generated, legally-binding equivalents of handwritten signatures. They uniquely identify the persons responsible for an action.

An electronic record is generated each time an action is signed. Electronic records consist of the name of the person(s) involved in the signing process. Electronic records are written to your SQL Server database and retained as a permanent record of a signed action. For more information on electronic records, see **Electronic Signature Tracking**.

When a task/task step action or form is configured to collect electronic signatures, the signatories must be the appropriate users. Only a user who is a member of the assigned group is able to sign.

Performing and Verifying Actions

Depending on how your forms are configured, a signed action may require a supervisor or another operator to verify or validate the action performed by the operator. The concept of "performed by" and "verified by" provides the foundation of understanding how electronic signatures work in Workflow.

An electronic signature is either a Perform Only signature or a Perform and Verify signature:

Perform Only Signature

The operator that performed the action must electronically sign for that action.

Perform and Verify Signature

The operator (that is, the performer) who performed the action must electronically sign for that action and another individual (the verifier) must electronically sign to verify that the action was performed. The electronic signature is not complete until both signatures are entered.



Note:

The person who performs an action cannot be the same person who verifies that action.

A signature consists of two components that uniquely identify the signer: a user name and a password. When the operator completes a form and clicks Sign Form, the electronic signature collection panel appears where the operator and, if applicable, the verifier must enter these two identifiers:

User Name

Login name of the user performing the action or verifying the action.

Password

Password for the user performing the action or verifying the action.



Note:

If an operator's Workflow user account was established by using Windows security, the Workflow user name and password are the same as the Windows user name and password. The user must also enter the Windows domain. However, on subsequent Windows account logins, the domain name is automatically entered.

An operator who performs or verifies an action can optionally enter a comment related to that action.

Related reference

[Electronic Signature Tracking \(on page 565\)](#)

Signatures During Workflow Run Time

When a running workflow requires an electronic signature for a particular section, the appropriate operator must sign and/or verify the task/task step action or form; otherwise, the workflow will not proceed further.

The need to sign for an action is determined by the way the task/task step action or form is configured by the workflow author.



Note:

A task/task step action or form can be cancelled without a signature if it has been preconfigured to do so; however, if you cancel, the task step is skipped and the workflow continues.

The electronic signature related to a completed task form can be viewed from **Completed Task Information**.

Compliance

The collected electronic signature data is stored in your SQL Server database in a separate table and can be used to generate reports for regulatory compliance.

Electronic Signature Tracking

Each time an operator signs an action, a detailed electronic record is written to the electronic signature audit trail.

Records written to the electronic signature audit trail:

- Ensure a tamper-resistant, time-stamped, permanent record of operator actions.
- Include the user names and full names of all performers and verifiers involved in signing and verifying actions.
- Include all comments entered by the performers and verifiers involved in signing and verifying actions.
- Include the unaltered description of the action signed for by the operator and as displayed in the **E-Signature Area** at the time of signing.
- Are recorded in your SQL Server database.

Electronic signature tracking starts when an operator clicks the Sign Form button on a form in a running workflow. An initial record is then created in the ElectronicSignature table in the SQL Server database with a state of Awaiting Performer. When the performer signature is completed, the record state is updated to Awaiting Verifier or Completed, depending on whether a verifier signature is required. The electronic signature related to a completed task form can be viewed from the **Completed Task Information**. If the signing is cancelled at any time, the state is updated to Cancelled and if the signing fails it is updated to Failed. For information on electronic signature failures, see **Failed Electronic Signature Examples**.

Audit records for electronic signatures are created at the conclusion of the electronic signature whether it completed successfully, was cancelled, or failed.

21 CFR Part 11 Overview

21 CFR Part 11 is a United States Government Food and Drug Administration (FDA)-mandated regulation that requires all electronic records and signatures, paperless records, and reporting procedures related to the manufacture of a product be captured and stored securely for businesses under its control, such as the Bio-Pharmaceutical and Food and Beverage industries.

This regulation requires the protection, accuracy, and quick retrieval of all records. Secured, computer-generated, time-stamped audit trails must be available to independently record the date and time of operator actions that modify the manufacturing process.

Electronic records can be used to identify the ingredients and people involved in the production and distribution of regulated substances, such as prescription drugs. Additionally, electronic records ensure accuracy, reliability, and security in data collection and record keeping.

Regulated industries that fail to meet 21 CFR Part 11 compliance risk the chance of Inspectional Observations (483s), warning letters, or the authorized shut-down of one or more operations.

The electronic signature option included with Workflow allows you to design a workflow and associated Form activities that assist you to meet the demands of this regulation. The paperless environment that results from using this feature benefits you with faster information exchange, improved ability to integrate, trend and search data, a reduction in errors, and reduced data storage costs.

GE Intelligent Platforms' 21 CFR Part 11 Services

GE Intelligent Platforms offers 21 CFR Part 11 consulting services to assist you with your goal of achieving 21 CFR Part 11 compliance. By using these services, you can reduce the time, effort, and expense of developing, implementing, and maintaining a compliant solution to meet the regulation.

These services include:

- Training
- Assessment
- Detailed Detection
- Maintenance

For more information, contact your GE Intelligent Platforms representative.

Audit Trails

Workflow provides audit trail capability that allows you to keep a traceable record of actions that are performed within your system. Actions that regulators are concerned with are typically those actions that commonly occur in a production environment.

The audit trail is a key component in a 21 CFR Part 11 compliant system, but it can also be useful in many different applications. The electronic signature audit trail contains a computer-generated, time-stamped record of each electronic signature. Each record clearly identifies all pertinent information about the person who entered the signature, such as the person's name, the time the signature was entered, and, optionally, comments provided by the signer.

Enabling a workflow may require auditing because that is when the workflow changes are entering the live process. Actions that are rarely performed can be tracked by using a manual paper system.

When an action that is configured to be audited takes place, an audit record is written to the SQL Server database. The audit entry in the database includes the following information:

- A time stamp indicating when the record was created.
- An optional time stamp indicating when the action took place.
- The location where the action took place (that is, the assigned equipment resource).
- User context, which is any information related to the audit record and can include a bind to a workflow variable and/or an expression.
- The associated topic.

Auditing can be enabled or disabled on a per topic basis; however, audit configuration changes and electronic signatures are always audited, regardless of whether the global auditing is enabled or disabled.

Configure Audit Trails and Electronic Signatures

The **Audit/E-Signature** editor is used to configure audit trails and electronic signature failure limits.

Related information

[Purpose of Electronic Signatures \(on page 562\)](#)

[Audit Trails \(on page 567\)](#)

Configure electronic signatures for use during workflow run time

You can configure electronic signatures to be available when a workflow is running to verify a task/task step action or a form.

Procedure

1. You can configure an electronic signature from various locations within a workflow depending on which action during run time you want to set an electronic signature against.

To add signature to a...	Do this...
Task	<ol style="list-style-type: none"> a. Open a workflow model resource. b. At the workflow level, click the E-Signature tab.
Task Step	<ol style="list-style-type: none"> a. Open a workflow model resource. b. Drag and drop a Local Subprocess into the workflow and select it, or select an existing subprocess. c. Click the E-Signature tab. d. Select the Override Workflow E-Signature check box.
Form	Configure a Form activity.

2. On the configuration panel, in the **Electronic Signature Settings** area, select the **Signature Required** check box.
3. In the **Valid Performers** field, click the ellipsis button to select the security group (personnel resource) that the signature performer must be a member of.
4. If you want the task/task step action or form to close and the workflow to proceed if the electronic signature fails, select the **Close if Signature Fails** check box.

**Note:**

A task/task step action or form can be cancelled without a signature if it has been preconfigured to do so; however, if you cancel, the task step is skipped and the workflow continues.

5. If a second signature is required before the task/task action or form can be submitted, do the following.
 - a. Select the **Verifier Required** check box.
 - b. In the **Valid Verifiers** field, click the ellipsis button to select the security group (personnel resource) that the signature verifier must be a member of.
6. In the **E-Signature Description** field, enter a description of the action being performed or the values that were entered on the task/task step action or form. This description is recorded with the electronic signature record.
7. Click **Save**.

Topics to be Audited

The **Audit Configuration** tab is used to select the topics that will be audited when auditing is enabled in your system.

Audit topics are grouped in related sets, called topic types, based on functional areas within the system. You can choose to audit all defined areas of your system or select individual functions in an area.

Auditing must be enabled for actions to be recorded in your database. The only exceptions are the Audit Configuration change and Sign Electronic Signature topics; these topics are audited and stored at all times.

Audit topics span all operations of the topic type. For example, the Equipment Model topic type includes a topic called Modify Property Value or Binding, which audits changes to all property values for all equipment resources. You cannot audit the changes made to an individual equipment instance (or class) property.

Audit Topics

The following lists describe the topics you can audit.

- Equipment Model topics
 - Add or Remove Property: Includes adding or removing instance and class properties.
 - Modify Equipment: Includes creating, deleting, and modifying equipment instance attributes, such as name and description.
 - Modify Equipment Class: Includes creating, deleting, and modifying equipment class attributes, such as name and description.
 - Modify Property Definition: Includes modifying equipment property attributes, such as name and description.
 - Modify Property Value or Binding:-Includes modifying the value or data binding of equipment properties.
- Event topics
 - Enable or Disable Event: Includes enabling or disabling time or condition events.
 - Import or Export Events: Includes importing and exporting event resources.
 - Modify Event: Includes creating, deleting, and modifying time or condition events.
- Material Model topics
 - Add or Remove Property: Includes adding or removing instance and class properties.
 - Modify Material Class: Includes creating, deleting, and modifying material class attributes, such as name and description.
 - Modify Material Definition: Includes creating, deleting, and modifying material definition attributes, such as name and description.
 - Modify Property: Includes modifying property attributes, such as name and description.
- Personnel Model topics
 - Add or Remove Property: Includes adding or removing instance and class properties.
 - Modify Person: Includes creating, deleting, and modifying person attributes.
 - Modify Personnel Class: Includes creating, deleting, and modifying personnel class attributes.
 - Modify Property: Includes modifying property attributes, such as name and description.
- Production Model topics
 - Modify Schedule Model: Includes modifying work requests, segment requirements, parameters, and resource requirements.
 - Modify Work Definition Model: Includes modifying work definition segment and work definition instance details, segment associations, parameters, and specifications.
 - Modify Work Process Segment Model: Includes modifying work process segments and work data classes details, parameters and properties, specifications, and work data associations.
- Regulatory Compliance

- Audit Configuration Change: Includes configuration changes to auditing.
- Sign Electronic Signature: Includes signing electronic signatures.
- Write audit Method: Includes entries generated by the WriteAuditEntry call method.
- Security topics
 - Authentication: Includes successful and failed authentication attempts.
 - Log In: Includes successful log-in attempts.
 - Log Out: Includes successful log-out attempts.
 - Modify Security Model: Includes creating, deleting, and modifying attributes of key sets, groups, and users.
 - Modify User Credentials: Includes changes to user credentials.
- Workflow topics
 - Enable or Disable Resource: Includes enabling and disabling workflow resources.
 - Import or Export Resources: Includes importing and exporting workflow resources.
 - Modify Task Status: Includes modifying task state, such as starting a task, cancelling a task, or terminating a task.
 - Task Step Jump: Includes task step jumping.

Related reference

[Failure Limits on Electronic Signatures \(on page 572\)](#)

[Configure Audit Trails and Electronic Signatures \(on page 567\)](#)

Related information

[Electronic Signatures and Audit Trails \(on page 562\)](#)

Configure audit trails

Before you begin

Procedure

1. In the navigator, click **Proficy System > Proficy System**.
2. In the **Displays** panel, click **Audit/E-Sig**.
3. Click the **Audit Configuration** tab.
4. From the topics lists, select the check box for each action you want to audit.

**Tip:**

To audit all areas of your system, select the **Select All Operations** check box.

5. Click **Save**.
6. Click **Enable**.

Failure Limits on Electronic Signatures

The **Electronic Signature Configuration** tab is used to configure the number of failed signing attempts before the electronic signature fails.

The configuration for failed signing attempts applies to the:

- Specific role that is configured for a form activity.
- Number of times a single user can attempt to sign an electronic signature instance before being locked out from successfully signing for one of the roles for that electronic signature instance.

The role for an electronic signature describes the role a user plays in the action being signed. There are two possible roles that a user can perform.

Performer

User who just performed the action.

Verifier

User who verifies the action performed and signed by the Performer.



Note:

You can assign a single group to both the Performer and Verifier roles or different groups to each role. Only members of those groups are permitted to electronically sign a form, but the same user cannot sign for both roles.

For added security, you can configure the number of times an electronic signature can be attempted on a single role before failing. For information and examples on signature failure, see **Failed Electronic Signature Examples**.

The following table describes the available attributes.

Attribute	Value	Description
Failure Limit on Role	User-defined	Specifies the number of times an electronic signature for the assigned role (that is, Performer or Verifier) can fail before the electronic signature fails.

Attribute	Value	Description
Failure Limit on Person	User-defined	Specifies the number of times a given user can attempt to sign a single electronic signature before the user is banned from signing in either the performer or verifier role for that electronic signature.

Related reference

[Failed Electronic Signature Examples \(on page 573\)](#)

Configure failure limits for electronic signatures

This procedure describes how to configure failure limits for electronic signatures.

Before you begin

Procedure

1. In the navigator, click **Proficiency System > Proficiency System**.
2. In the **Displays** panel, click **Audit/E-Sig**.
3. Click the **Electronic Signature Configuration** tab.
4. In the **Failure Limit on Role** field, enter the number of times that an incorrect electronic signature for a specific role can be entered before it fails.
5. In the **Failure Limit on Person** field, enter the number of times that an incorrect electronic signature for a specific user can be entered before the user is banned from signing in either the performer or verifier role for that specific signature.
6. Click **Save**.

Failed Electronic Signature Examples

Electronic signatures are configured with failure limits for added security. Failure limits prevent an unauthorized user from signing a form, or prevent an authorized user to enter incorrect credentials beyond the configured limit.

The role limit is the number of permitted attempts to provide a valid Performer or Verifier signature. Therefore, failures are on a per role basis.

The person (user) limit is the number of times a user can attempt to sign a form in the capacity of the specified role.

**Note:**

All signing attempts are based on a form instance.

The following examples describe the different ways that an electronic signature can fail, based on the configuration of your failure limits for roles and users.

Example A: Electronic signature fails when person limit is reached

- Role Limit = 4
- Person Limit = 2

Scenario	Result
User1 enters incorrect signature credentials on the first attempt.	Signing fails.
User1 enters incorrect signature credentials on the second attempt.	Signing fails and the user is locked out from signing this form instance.
User1 enters correct signature credentials on the third attempt.	Signing fails due to the user being locked out from signing this form instance.

Example B: Electronic signature fails when role limit is reached

- Role Limit = 3
- Person Limit = 2

**Note:**

Validation errors do not count against the configured failure limits. For example, if the user name is initially entered incorrectly, it can be corrected before signing the form. This error does not count against the failure limit because the form was not signed.

Invalid signing attempts by a user count towards the role limit until that user is locked out. When a user is locked out, no entry for that user (valid or invalid) is counted towards the role limit.

Scenario	Result
User1 enters an incorrect password.	Signing fails.

Scenario	Result
User2 is not a member of the assigned group (Valid Performers or Valid Verifiers).	Signing fails.
User3 enters an incorrect user name.	Signing fails and the electronic signature fails because the role limit has been reached.

Example C: Electronic signature fails when role limit is reached

- Role Limit = 3
- User Limit = 6

Electronic signature failures are tracked in the SQL Server database in both the ElectronicSignature and AuditRecord tables. For more information on electronic signature audit records, see **Electronic Signature Tracking**.

Scenario	Result
User1 enters incorrect signature credentials three times. On the fourth attempt (with valid or invalid credentials) a message appears stating that the user is locked out.	Signing fails; however, User1 can continue to attempt to log in with either valid or invalid credentials and those attempts do not count towards the role limit.
User2 enters incorrect signature credentials three times. On the fourth attempt (with valid or invalid credentials), a message appears stating that the electronic signature has failed.	Signing fails and the electronic signature fails because the role limit has been reached.

Related reference

[Electronic Signature Tracking \(on page 565\)](#)

Audit Trail Reports

Workflow provides the ability to generate reports based on your audit trail and electronic signature data.



Note:

You must have the Proficy Reporting Database installed and running in order to generate audit trail and electronic signature reports. For more information, see **Reporting Database and Reporting**.

Audit trail and electronic signature reporting uses the reporting database, not the production database. That is, in order for a record to appear in an audit trail report, the reporting database must have been synchronized with the production database since the record was logged. For information on synchronizing the reporting and production databases, see **Default Reporting Schedule Editor**.

Reports are generated using SQL Server Reporting Services (SSRS) and are available via a web page hosted by Internet Information Services (IIS). Reports can also be hosted in the Proficy Client by creating a user display that is a web page. For more information, see **User Displays**.

All reports are sorted chronologically and have the ability to filter over a start and end date range. You can perform additional filtering to streamline your reports. For example, you can filter by audit type, audit topic, location, or user name.

You can export generated reports into various file formats. These file formats include:

- XML
- CSV
- TIFF
- PDF
- Web archive
- Excel



Note:

When you export a report, only the Web archive and Excel formats maintain the hyperlink(s) to the other reports. Access to the reporting server is needed to link to other reports.

Report Types

Audit Trail Report

The audit report contains audit information and a link to electronic signature data, if an action is related to a signature. You can use any combination of the query fields to filter the report data.

The audit trail report includes the following information:

Column	Description
Record Start Date	Filter by the earliest date for collection of report information.

Column	Description
Record End Date	Filter by the latest date for collection of report information.
Audit Type	Filter reports by report type.
Audit Topic	Filter reports by the topic name.
Record Time	Filter by the time the record of the action was created.
Message	Filter by the message on the recorded actions. You can filter reports by searching for message contents. Wildcard searches using the % character are supported.
Location	Filter by the name of the Client computer that generated the report content. Wildcard searches using the % character are supported.
Context Time	The time the recorded actions occurred. This is recorded only when there is a difference from the Record Time. Wildcard searches using the % character are supported.
Username	Filter by the user name of the performer for the recorded actions. Wildcard searches using the % character are supported.
Context	Additional context added to the audit record. Wildcard searches using the % character are supported.

Electronic Signature Report

The electronic signature report contains information on the action performer and verifier (if there is a verifier).

Column	Description
Start Date	Filter by the earliest date for collection of report information.

Column	Description
End Date	Filter by the latest date for collection of report information.
Status	Filter by the status of the electronic signing: <ul style="list-style-type: none"> • successful/completed • failed • cancelled • awaiting performer • awaiting verifier <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: To filter by awaiting performer and awaiting verifier statuses, search for not successful, not cancelled, and not failed status reports. </div>
Is Successful	Filter by successful or not successful.
Is Cancelled	Filter by cancelled or not cancelled.
Is Failed	Filter by failed or not failed.
Performer Group	Filter by the personnel group of the performer.
Verifier Group	Filter by the personnel group of the verifier.
Performer Username	Filter by the user name of the performer.
Verifier Username	Filter by the user name of the verifier.
Performer Full Name	Filter by the full name of the performer.
Verifier Full Name	Filter by the full name of the verifier.
Performer Comment	Filter by the comment content of the performer.
Verifier Comment	Filter by the comment content of the verifier.
Performer Signing Time	Filter by the performer's signing time.
Verifier Signing Time	Filter by the verifier's signing time.

Sample Reports

The sample reports are found under your Workflow installation folder: `../Proficy/Proficy Platform Reporting/Sample Reports/SOAAuditingReports`.

The sample reports must be loaded into the reporting database to be accessed.

Report	Description
ErrorDetails.rdl	Provides execution errors if the data synchronization fails.
ExecutionDetails.rdl	Provides execution detail of each data synchronization, including package execution result, quantity synchronized, and so on.
ExecutionHistory.rdl	Provides the history of the data synchronization.
SOAAuditingReports.rptproj	The reporting project file.
SOAReportingDB.rds	The reporting data source configuration.

Add the sample audit reports

Before you begin

The reporting database must be installed before you can proceed.

About this task

For more information, see the SQL Server Reporting Services web interface documentation.

Procedure

1. In the SQL Server Reporting Services web interface, add the reporting database as a new data source.
2. Browse to the sample reports location (`../Proficy/Proficy Platform Reporting/Sample Reports`) and select a report file (.rdl) to upload.
3. Set the data source of the loaded report to the reporting database.

View electronic signature and audit reports

Before you begin

The reporting database must be installed before you can proceed.

Procedure

1. In the **SQL Server Reporting Services** web interface, upload the report file you want to use.
2. In **Global Displays**, click **Displays**.
3. In the navigator, click **Global Displays**.
4. To add a display group, in the **Folders** list, click **Displays**, click the arrow next to **Add Display**, and then click **Add Display Group**.
5. To add a forms group, in the **Folders** list, click **Forms**, click the arrow next to **Add Form**, and then click **Add Form Group**.
6. In the **Name** and **Description** fields, enter a name and description for the resource.
7. Click **OK**.
8. In the **Display Editor**, select **Web Page** and enter the URL of the uploaded report file.
9. Click **Save**.
10. In the **Show Compatible** panel, click **Global Viewer**.
The reporting web interface appears.

Synchronize the run-time and reporting databases

Before you begin

The reporting database must be installed before you can proceed.

About this task

For information on synchronizing the run-time and reporting databases, see Default Reporting Schedule Editor.

Filter audit reports

Before you begin

The reporting database must be installed before you can proceed.

Procedure

1. In the **SQL Server Reporting Services** web interface, click the **View** tab.
2. To set the time range of audit reports, in the **Record Start Date** and **Record End Date** fields, enter the date and time.
3. To filter reports, from the **Audit Type** and **Audit Topic** drop-down lists, select an item or enter search criteria in one of the following fields.
 - Message
 - Username
 - Location
 - Context

**Note:**

Wildcard searches using the % character are supported. For example, you can search for "topic % modified" in the Message field and get all reports where the message included entries such as: topic modified, topic configuration has been modified, and topic events modified.

Filter electronic signatures

Before you begin

The reporting database must be installed before you can proceed.

Procedure

1. In the **SQL Server Reporting Services** web interface, click the **View** tab.
2. To set the time range of electronic signatures, in the **Start Date** and **End Date** fields, enter the date and time.
3. To filter electronic signatures, from one of the following drop-down lists, select an item:
 - Performer Groups
 - Verifier Groups
 - Is Successful
 - Is Cancelled
 - Is Failed

Or, in one of the following fields, enter search criteria.

- Performer Username
- Verifier Username
- Performer Full Name
- Verifier Full Name
- Performer Comment
- Verifier Comment

**Note:**

Wildcard searches using the % character are supported. For example, you can search for % in the Verifier Username field and get all reports where a verifier signature was made.

IAuditTrail

IAuditTrail allows you to write entries to the audit trail.

On the activity configuration panel, you can use a bind or expression to define your input and output parameters.

Subtopics in this section describe each service method for this service interface.

Data Sources

Data Sources

Workflow provides the ability to connect to external data sources to access the data contained within those applications.

When these data sources are configured, their data (for example, tags, displays, SQL statements) are used to bind to data items in equipment, workflows, and condition events. The available data sources and their uses are as follows:

- The Historian data source allows you to connect to Historian servers to read from and write to Historian data tags. You add Historian tags to Workflow and then use those tags as data sources throughout Workflow.
- An OPC client provides direct access to data served by OPC Data Access (DA) servers without relying on Historian as a bridge between an OPC server and Workflow.
- An OPC UA client provides direct access to data served by OPC Unified Architecture (UA).

The Data Source Editor is used to select the data tags and other data items to be used when configuring data items, as well as when configuring connections to Historian servers, OPC Clients, and OPC UA Clients.



Note:

- You can add up to 3,000 data items from each Historian and OPC DA data source you connect to.
- You can add up to 5,000 data items from each OPC UA data source you connect to.
- The Historian server you want to connect to must be on the same network domain or workgroup as the Workflow server.
- You can configure the Historian server only once.

For information on working with data items, see **Data Items**.

Configure a data source item for a resource

This procedure is used to configure data source items for equipment, events, and workflows.

Procedure

1. From the navigator, select the model that contains the resource you want to work with.
In the next pane, a list of resources appears.
2. From this list, select the resource you want to work with.
3. If applicable, from the next list in the hierarchy, select the resource you want to work with.



Tip:

As more lists appear, select the appropriate resource until you find the one you are looking for.

4. Access the **Configure Data Source** dialog box from within the current editor.

For...	Access the...
Equipment	Properties tab.
Condition Events	Variables tab.
Workflows	Edit Data Items dialog box.

5. Expand the data source that contains the data item you want to bind to, and then select the data item.
6. Click **Next**.
7. In the Item specific configuration section, configure the method.



Note:

Item configuration depends on the selected source and item. Some items do not require configuration.

8. At run time, if you want to return all data from the selected item, select the Return all data option.
9. At run time, if you want to return a subset of data from the selected item at run time, select the Return only a subset of data option, and then from the data tree, select the data you want returned.

**Note:**

If the item returns only a single piece of data, the Return all data option is selected, and you cannot retrieve a subset of data.

10. Click **Finish**.

GE Historian Data Source

The GE Historian data source allows you to connect to Historian servers to read from and write to Historian data tags. You add Historian tags to Workflow and then use those tags as data sources throughout Workflow.

**Note:**

Workflow supports GE Historian 4.5 or greater.

Historian server names and display names are distinct from each other. A GE Historian display name can contain spaces and/or other special characters. A server name is the address of the GE Historian server and must comply with a valid Windows@computer name; however, the underscore character is permitted.

Add Historian Server

The following table describes the attributes that are available when adding a new GE Historian server.

Attribute	Value	Description
Display Name	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 255 characters. An alias can contain spaces and special characters. A server address can contain the underscore character.	Specifies the GE Historian server name as you want it to appear in the navigator. This name can be the server address or an alias.
Description	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 255 characters.	Specifies optional information about the GE Historian server.
Server Name	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 255 characters. A server address can contain the underscore character.	Specifies the actual address/computer name of the GE Historian server.

Attribute	Value	Description
		You can also enter or update this field on the Data Source Editor.
User Name	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 255 characters.	<p>Specifies the user name used to connect to the GE Historian server. Workflow uses this name when retrieving data from the GE Historian server.</p> <div data-bbox="1027 625 1419 890" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;">  Note: This field is required when security is enabled on the GE Historian server. </div> <p>You can also enter or update this field on the Data Source Editor.</p>
Password	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 255 characters.	<p>Specifies the password used to connect to the GE Historian server. Workflow uses this password when retrieving data from this server.</p> <div data-bbox="1027 1276 1419 1541" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;">  Note: This field is required when security is enabled on the GE Historian server. </div> <p>The password is not displayed in this field for security reasons. When you enter a password, an asterisk (*) displays for every character you specify.</p>

Attribute	Value	Description
		You can also enter or update this field on the Data Source Editor.
Domain	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 255 characters.	<p>Specifies the domain for the GE Historian server. Workflow uses this domain name when retrieving data from the GE Historian server.</p> <div data-bbox="1024 621 1419 890" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: This field is required when security is enabled on the GE Historian server. </div> <p>You can also enter or update this field on the Data Source Editor.</p>

GE Historian Details

The following table describes the GE Historian server attributes in the **Data Source Editor**:

Attribute	Value	Description
Description	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 255 characters	Provides optional information about the GE Historian server.
Server Name	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 255 characters	Specifies the address of the GE Historian server that Workflow is connected to.
User Name	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 255 characters	Specifies the user name used to connect to the GE Historian server. Workflow uses this name when retrieving data from the GE Historian server.

Attribute	Value	Description
		 Note: This field is required when security is enabled on the GE Historian server.
Password	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 255 characters	Specifies the password used to connect to the GE Historian server. Workflow uses this password when retrieving data from this server.  Note: This field is required when security is enabled on the GE Historian server. The password is not displayed in this field for security reasons. When you enter a password, an asterisk (*) displays for every character you specify. You can also enter or update this field on the Data Source Editor.
Domain	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 255 characters	Specifies the domain for the GE Historian server. Workflow uses this domain name when retrieving data from the GE Historian server.  Note: This field is required when security is enabled

Attribute	Value	Description
		<div data-bbox="1024 268 1419 373" style="border: 1px solid #0070C0; border-radius: 10px; padding: 5px; display: inline-block;">  on the GE Historian server. </div> <p data-bbox="1024 411 1419 489">You can also enter or update this field on the Data Source Editor.</p>

Tag Information

The following table describes the areas that allow you to view information about the GE Historian tags found in Workflow:

Item	Description	
Start Synchronization	Click to browse the current tag status and event details available on the selected GE Historian server. This button can also be used to perform a manual re-synchronization of all tags in GE Historian with the tags visible in Workflow; however, this operation is normally performed automatically by Workflow.	
View Tags	Click to open the Historian Tags dialog box to view the existing tags on the selected GE Historian server.	
Remove Orphan Tags	Click to open the Orphan Tags dialog box and remove existing tags from Workflow that are no longer being accessed. When a tag is deleted from the GE Historian server, it is not automatically removed from Workflow; it is marked as orphaned so that ex-	

Item	Description	
		isting data links to that tag are not broken.

View Tags

The following table describes the actions available in the **Historian Tags** dialog box:

Item	Description	
Results		Displays the list and quantity of GE Historian tags that are currently available in Workflow. The number of pages of results returned is below this list.
Filter By		Enter text to filter the tags list to a specific list of tags, or to look for a specific tag.
Refresh		Click to activate the filter.

Remove Orphan Tags

The following table describes the actions available in the **Orphan Tags** dialog box:

Item	Description	
Results		Displays the list and quantity of GE Historian tags that were not found on the GE Historian server during the last tag synchronization. The number of pages of results returned is below this list.
Filter By		Enter text to filter the tags list to a specific list of tags, or to look for a specific tag.
Refresh		Click to activate the filter.

Item	Description	
Remove All	Click to remove all the tags from the list.	
Remove	Click to remove a selected tag from the list.	

Add a GE Historian server

You can add an GE Historian server as a data source.

Before you begin



Note:

Workflow supports GE Historian 4.5 or greater.

Procedure

1. In the navigator, click **Proficy System > Proficy System > Historian**.
2. Click **Add Historian**.
3. In the **Display Name** field, enter either the actual name of the GE Historian server you want to connect to or an alias for the server name.
The name you enter in this field is automatically entered into the **Server Name** field.
4. If you entered an alias in the **Display Name** field, then in the **Server Name** field, enter the GE Historian server address.



Note:

The GE Historian server address must comply with a valid Windows[®] computer name; however, the underscore character is permitted.

5. **Optional:** Enter a description of the GE Historian server.
6. If security is enabled on the GE Historian server, enter the user name, password, and domain for this server.
7. Click **OK**.

Change the GE Historian server

You can connect to a new GE Historian server at any time and maintain all the links to equipment properties, condition events, and workflow data items. You retain the alias for the server but change the server address.

Before you begin

To prevent links between GE Historian tags and equipment properties, condition events, and workflow data items from breaking when you change servers, you must make sure that all the tags that exist on the original GE Historian server also exist on the server you are switching to. If the original server has tags that do not exist on the new server, those tags will not be available when you switch to the new server and synchronize the tags.

About this task**Note:**

- Workflow supports GE Historian 4.5 or greater.
- You cannot change the alias of your GE Historian server, even when you change servers. To change the alias, you must delete the connection and add a new one.

Procedure

1. In the navigator, click **Proficiency System > Proficiency System > Historian**.
2. From the **Historian Servers** list, select the display name of the server that you want to change.
3. From the **Displays** panel, select **HW/SW > Data Source Editor**.
The **Data Source Editor** appears.
4. In the **Server Name** field, enter the address of the GE Historian server you want to change to.
5. If security is enabled on the GE Historian server, enter the user name, password, and domain for this server.
6. Click **Save**.

View and synchronize GE Historian tags

You can browse the status of current tag and event details available on the GE Historian server, as well as resynchronize all server tags with client tags.

Before you begin**Note:**

Workflow supports GE Historian 4.5 or greater.

Procedure

1. In the navigator, click **Proficiency System > Proficiency System > Historian**.
2. From the **Historian Servers** list, select the server whose tags you want to view and/or synchronize.
3. From the **Displays** panel, select **HW/SW > Data Source Editor**.

The **Data Source Editor** appears.

4. To browse the current tags in Workflow, click **View Tags**.

This **Historian Tags** dialog box appears.

5. **Optional:** To reduce the list of tags or to search for a specific tag, use the **Filter By** field and **Refresh** button.
6. Click **Close**.
7. To manually resynchronize GE Historian tags, click **Start Synchronization**.
8. Click **Save**.

Remove orphaned GE Historian tags from Workflow

Before you begin



Note:

Workflow supports GE Historian 4.5 or greater.

Procedure

1. In the navigator, click **Proficy System > Proficy System > Historian**.
2. From the **Historian Servers** list, select the server whose orphaned tags you want to remove.
3. From the **Displays** panel, select **HW/SW > Data Source Editor**.
The **Data Source Editor** appears.
4. In the **Tag Information** section, click **Remove Orphan Tags**.
The **Orphan Tags** dialog box appears.
5. To reduce the list of tags or to search for a specific tag, use the **Filter By** field and **Refresh** button.
6. Click **Remove** to remove the selected tag or **Remove All** to remove all orphaned tags.
7. Click **Save**.

GE Historian Synchronization Tips

The following information provides guidelines for improving the performance of your GE Historian synchronization process.

Indexes on the Historian_Tag table

If the indexes are badly fragmented due to many deletes and/or inserts, which can occur if the synchronization has been attempted multiple times, the process will be slow. You can rebuild the indexes using SQL Server Management Studio as follows.

1. Expand the **Historian_Tag** table.
2. Right-click the **Indexes** folder, and then select **Rebuild All**.

Rebuilding the indexes should be performed before you attempt to synchronize. It should also be performed as a maintenance operation after the initial synchronization so that the table is in a good state for later synchronizations.

Memory on SQL Server

The initial synchronization is memory-intensive. The more RAM available to the SQL Server, the faster it will be. Ideally, SQL Server is on a dedicated computer. You should also ensure that SQL Server is configured with an appropriate **Maximum server memory** setting and not the default value, which is effectively infinite. For more information, see <http://www.sqlchicken.com/2012/02/setting-memory-limits/>.

SQL server recovery model

- Testing has shown that if the recovery model for the SOADB is set to *Simple* vs. *Full*, database operations complete faster. However, there are other considerations with regard to data recovery that you must be aware of. For more information, see <https://msdn.microsoft.com/en-us/library/ms189275.aspx>.

Setting SOA to manual synchronization

A setting can be added to the `ProficiencyServer.exe.config` file that prevents automatic synchronization when Workflow Server starts. With large systems, manual mode can be used so that the synchronization process is triggered only when it is explicitly initiated.

To prevent automatic synchronization, under the `<appSettings>` section, add the entry `<add key="HistorianManualSyncOnly" value="true" />`.

OPC Client Data Source

An OPC client provides direct access to data served by OPC Data Access (DA) servers without relying on GE Historian as a bridge between an OPC server and Workflow.



Note:

Supported OPC DA specifications are: v2.0x and 3.0.



Tip:

You can add up to 3000 groups to each OPC server and 5000 items per group.

OPC items added to Workflow can be used as data items for workflows, events, and equipment properties. Workflow can read from and write to these data items. OPC servers configurations can be imported and exported to other systems.



Important:

- Workflow supports local OPC servers only; it does not support remote OPC servers.
- The OPC client supports synchronous reads and writes only; it does not support asynchronous reads or writes.

Edit Description

You can change the description of this resource.

Attribute	Value	Description
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

The following table describes the OPC Server attributes on the Data Source Editor:

Attribute	Value	Description
OPC Server Name	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 255 characters	Displays the name of the OPC server.
Description	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 255 characters	Provides more information that a user may find helpful.
Server Prog ID	User-defined	Specifies the ProgID of the OPC server you want to connect to. You can select from the list or enter the ProgID. Refer to your OPC server documentation to determine the ProgID.

Attribute	Value	Description
Health Monitor Frequency	User-defined numeric value	<p>Specifies how often the OPC server communicates its health status to the OPC Client.</p> <div data-bbox="1026 432 1419 1220" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note:</p> <ul style="list-style-type: none"> • If you are using OPC DA v2.0x, the client communicates with the server at this interval to monitor the health status. • If you are using OPC DA v3.0, the server communicates with the client at this interval to monitor the health status. </div>
OPC Server Status	N/A	<p>Displays the status of the OPC server. Status information includes:</p> <ul style="list-style-type: none"> • Connection Status: connected or not connected • State: running or not running • Status Info: running normally or not • Vendor Info: The name and web site of the OPC Server vendor company

Attribute	Value	Description
		<ul style="list-style-type: none"> • Product Version: the version number of the OPC Server • Start Local Time: the date and local time the server was started • Current Local Time: the date and local time the server status was refreshed • Last Update Local Time: the date and local time the server was last updated <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: The OPC server status information is not automatically refreshed; you must refresh the status manually. </div>
Refresh	N/A	Click to refresh the server status information.

Add OPC Items to Workflow

Before you can add OPC items to Workflow, you must add one or more groups to the OPC server. You can choose to detect changes in an OPC item value by enabling the associated groups, or by reading the OPC data item. If a group is not enabled, no event notifications about data changes are sent to subscribers of the associated items.

You can also choose to configure a group to read OPC items from the OPC server's cache, or to get values directly from the device the server is reading from.

After you add your OPC groups, you can browse the OPC server and add up to 5000 items to each group. The items are automatically added to Workflow when you add them to a group. If you want to add only a subset of the available items, you can expand the top level item list in the Server Items area and then

select the check box beside each item or set of items you want to add. You can also remove items from the Selected Items list

The following table describes the OPC Server Group attributes on the Data Source Editor.

Attribute	Value	Description
OPC Group Name	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 255 characters	Displays the name of the OPC Group.
Description	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 255 characters	Provides more information that a user may find helpful.
Update Rate	User-defined	Specifies the rate, in milliseconds, at which the server is requested to update items in this group. Event notifications are sent to subscribers of items associated with the selected group when data changes. The default rate is 1000 ms.
Enabled	User-defined	Select this check box to allow event notifications to be sent to subscribers of items associated with the group. Event notifications are sent when data changes and at the rate specified in the Update Rate field. Groups are enabled by default.
Read Mode	System-defined	<p>Select the method to use for reading OPC item values.</p> <ul style="list-style-type: none"> • Cached - OPC item values are read from the OPC server's cache. • Device - the OPC server bypasses its cache and

Attribute	Value	Description
		reads the OPC item values directly from the device the server is communicating with.
Browse Server	N/A	Click to browse the selected server for items.
Server Items	N/A	Displays the items contained by the selected server. Select the highest level check box to select all of the items and add them to the Selected Items list. This check box is not available if the server contains more than 5000 items.
Selected Items	N/A	Displays the items you selected to add to Workflow.
Remove	N/A	Click to remove the selected item from Workflow. Removing an item from Workflow does not remove it from the server.

Add an OPC client

You connect your OPC client to an OPC server that is installed on your application server computer.

Procedure

1. In the navigator, click **Proficy System > Proficy System > OpcClient**.
2. Click **Add OpcServer**.
3. In the **Name** field, enter a name for the OPC server you want to connect to.
4. Optionally, in the **Description** field, enter a description of the OPC server.
5. In the **Server Prog ID** field, enter the Prog ID of the OPC server you want to connect to.

- Or -

Click the ellipsis button to open the **Select OPC Server Prog ID** dialog box where you can select the OPC server you want to connect to.

Refer to your OPC server documentation to determine the Prog ID.

6. Click **OK**.

Configure the OPC server

You configure an OPC server to ensure it is connected and its status is regularly communicated to your OPC client.

Before you begin

Add an OPC client.

Prerequisites

[Add an OPC client \(on page 598\)](#)

Procedure

1. In the navigator, click **Proficy System > Proficy System > OpcClient**.
2. From the **OPC Servers** list, select the OPC server you want to configure.
3. From the **Displays** panel, select **HW/SW > Data Source Editor**.
The **Data Source Editor** appears.
4. If applicable, select a new server Prog ID:
 - a. Click the ellipsis button next to the **Server Prog ID** field.
The **Select OPC Server Prog ID** dialog box appears.
 - b. Select the Prog ID of the OPC server you want to connect to, and then click **OK**.
5. In the **Health Monitor Frequency** field, enter how often, in milliseconds, the OPC server communicates its health status to the OPC client.



Note:

- If you are using OPC DA v2.0x, the client communicates with the server at this interval to monitor the health status.
- If you are using OPC DA v3.0, the server communicates with the client at this interval to monitor the health status.

6. Click **Refresh**.

The OPC server status information is displayed.



Note:

The OPC server status information is not automatically refreshed; you must refresh the status manually.

7. Click **Save**.

Add groups to the OPC client

You must add groups to your OPC client in order to browse the items that are stored on the OPC server.

Before you begin

Configure the OPC server.

Prerequisites

[Configure the OPC server \(on page 599\)](#)

Procedure

1. In the navigator, click **Proficy System > Proficy System > OpcClient**.
2. In **OPC Servers** list, select an OPC server.
3. Click **Add OpcGroup**.
4. In the **Name** field, enter a name for the OPC group.
5. Optionally, in the **Description** field, enter a description of the OPC group.
6. Click **OK**.

Add OPC items to Workflow

Procedure

1. In the navigator, click **Proficy System > Proficy System > OpcClient**.
2. In **OPC Servers** list, select an OPC server.
3. From the **OPC Groups** list, select the group you want to browse.
4. From the **Displays** panel, select **HW/SW > Data Source Editor**.
The **Data Source Editor** appears.
5. If applicable, in the **Update Rate** field, enter the rate at which the server is requested to update items in this group.
6. Select the **Enabled** check box to allow event notifications to be sent to subscribers of items associated with the group.

7. From the **Read Mode** drop-down list, select whether the OPC item values are read from the OPC server's cache or directly from the device the server is communicating with.
8. In the **Server Items** section, click **Browse Server**.
9. To add individual items or continuous groups of items, expand the list of items you want to select from, and then select the check box beside each item you want to add to Workflow.
As you select each check box, the item name appears in the **Selected Items** list.

**Tip:**

To add all of the items, select the highest level check box. This check box is available only if the server contains 5000 or fewer items.

10. Click **Save**.

Remove OPC items

Procedure

1. In the navigator, click **Proficy System > Proficy System > OpcClient**.
2. In **OPC Servers**, click the OPC server you want to work with.
3. From the **OPC Groups** list, select the group with the items you want to remove.
4. From the **Displays** panel, select **HW/SW > Data Source Editor**.
The **Data Source Editor** appears.
5. From the **Selected Items** list, select the item(s) you want to remove, and then click Remove.

**Note:**

Use **Shift+click** to select contiguous items or **Ctrl+click** to select individual items.

6. Click **Save**.

OPC UA Client Data Source

The OPC Unified Architecture (UA) client provides direct access to data served by OPC UA servers.

OPC UA items added to Workflow can be used as data items for workflows, events, and equipment properties. Workflow can read from and write to these data items. OPC UA server configurations can be imported and exported to other systems.

**Tip:**

You can add up to 5,000 items per group.

Edit Description

You can change the description of this resource.

Attribute	Value	Description
Description	User defined up to 255 characters.	Provides more information about a resource that a user may find helpful.

The following table describes the OPC UA server attributes on the **OPC UA Editor**:

Attribute	Value	Description
OPC UA Server Name	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 255 characters.	Displays the name of the OPC UA server.
Description	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 255 characters.	Provides more information that a user may find helpful.
URL	User defined.	Specifies the URL of the OPC UA server you want to connect to. You can select from the list or enter the URL. Refer to your OPC UA server documentation to determine the URL.
Check Connection	N/A	Displays the status of the OPC UA server. Status information includes: Connection Status: connected or not connected

Please see **Authentication** for more information about authentication settings.

Add OPC UA Items to Workflow

Before you can add OPC UA items to Workflow, you must add one or more item groups to the OPC UA server. Item groups allow OPC UA data items to be organized for easier navigation.

After you add your OPC UA group(s), you can browse the OPC UA server and add up to 5000 items to each group. The items are automatically added to Workflow when you add them to a group. If you want to add only a subset of the available items, you can expand the top-level item list in the **Server** items area, and then select the check box beside each item or set of items you want to add. You can also remove items from the **Selected Items** list.

The following table describes the OPC UA Server Group attributes on the **OPC UA Editor**:

Attribute	Value	Description
OpcUaGroup Name	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 255 characters.	Displays the OPC UA Group.
Description	User-defined alphanumeric string (A–Z, a–z, 0–9) up to 255 characters.	Provides more information that a user may find helpful.
Browse Server		Click to browse the selected server for items.
Server Items	N/A	Displays the items contained by the selected server. Select the highest-level check box to select all of the items and add them to the Selected Items list. This check box is not available if the server contains more than 5000 items.
Selected Items	N/A	Displays the items you selected to add to Workflow.
Remove	N/A	Click to remove the selected item from Workflow. Removing an item from Workflow does not remove it from the server.

Add an OPC UA Server

You can connect to an OPC UA server that is installed on your application server computer.

About this task

Before you can add an OPC UA Server, you will need to generate a Workflow certificate and register it with the GDS server. For more information, see **Register Workflow Certificates with Global Discovery Server (GDS)**.

Procedure

1. In the navigator, click **Proficy System > Proficy System > OPC UA**.
2. Click **Add OPC UA Server**.
3. In the **Name** field, enter a name for the OPC UA server to which you want to connect.
4. **Optional:** In the **Description** field, enter a description of the OPC UA server.
5. Click **OK**.

Configure an OPC UA Server

You can configure an OPC UA server to ensure it is connected and its status is regularly communicated.

Procedure

1. In the navigator, click **Proficy System > Proficy System > OPC UA**.
2. From the **OPC UA Servers** list, select the OPC UA server you want to configure.
3. From the **Displays** panel, select **HW/SW > OPC UA Editor**.

The **OPC UA Editor** appears.

4. If applicable, select a new server URL:
 - a. Click the ellipsis (...) next to the **Server URL** field.
The **OPC UA Server Details** dialog box appears.
 - b. Enter the `URL` of an OPC UA discovery server, and then click **Discover**.
 - c. Select the new **URL** in the **Server Name** field.
The Endpoints for the selected OPC UA server appear.
 - d. Select an endpoint from the list, and then click **OK**.
Alternatively, enter the desired OPC UA server URL directly.

**Tip:**

If a **SignAndEncrypt** endpoint is selected, the **Use Security** check box in the **OPC UA Server Details** dialog box is automatically checked.

5. In the **Authentication Settings**, choose from the following:

- **Anonymous** - used if the endpoint selected has no security and an anonymous login is allowed.
- **User/Password** - used if the endpoint selected has security, or an anonymous login is not allowed.

If **User/Password** is chosen, enter the `User Name` and `Password` for the selected server.

6. Click **Save**, and then click **Check Connection**.

The system displays a green **Connected** message.



Note:

If a red **Disconnected** message appears, check your authentication settings.

Add Groups to OPC UA

You must add groups to your OPC UA client in order to browse and select the items that are stored on the OPC UA server.

Procedure

1. In the navigator, click **Proficy System > Proficy System > OPC UA**.
2. In the **OPC UA Servers** list, select an OPC UA server.
3. Click **Add Item Group**.
4. In the **Name** field, enter a name for the OPC UA group.
5. **Optional:** In the **Description** field, enter a description of the OPC UA group.
6. Click **OK**.

Add OPC UA Items to Workflow

You can add asset tags from the OPC UA server to Workflow.

Procedure

1. In the navigator, click **Proficy System > Proficy System > OPC UA**.
2. In the **OPC UA Servers** list, select the OPC UA server you want to work with.
3. From the **OPC UA Groups** list, select the group you want to browse.
4. From the **Displays** panel, select **HW/SW > OPC UA Editor**.
The **OPC UA Editor** appears.
5. Click **Browse Server**.
A list of server items appears.
6. To add individual items or contiguous groups of items, expand the list of items you want to select from, and then select the check box beside each item you want to add to Workflow.
As you select each check box, the item name appears in the **Selected Items** list.

**Note:**

To add all of the items, select the highest-level check box. This check box is available only if the server contains 5000 or fewer items.

7. Click **Save**.

Remove OPC UA Items from Workflow

You can remove asset tags that were added to Workflow from the OPC UA server.

Procedure

1. In the navigator, click **Proficy System > Proficy System > OPC UA**.
2. In the **OPC UA Servers** list, select the OPC UA server you want to work with.
3. From the **OPC UA Groups** list, select the group with the items you want to remove.
4. From the **Displays** panel, select **HW/SW > OPC UA Editor**.
The **OPC UA Editor** appears.
5. From the **Selected Items** list, select the item(s) you want to remove, and then click **Remove**.

**Tip:**

Use `Shift+click` to select contiguous items or `Ctrl+click` to select individual items.

The items are removed from Workflow but are not removed from the data source server.

6. Click **Save**.

Set OPC UA Subscription Parameters (Advanced)

You can customize subscription parameters for an OPC UA server connection.

About this task

See OPC UA documentation for details on how to use the parameters. In most cases, the default values are sufficient.

Procedure

1. In the navigator, click **Proficy System > Proficy System > OPC UA**.
2. In the **OPC UA Servers** list, select the OPC UA server you want to work with.
3. From the **Displays** panel, select **HW/SW > OPC UA Editor**.
The **OPC UA Editor** appears.
4. Enter values for the following subscription parameters:

Value field	Description	Default Value
Operation Timeout	How much time is allowed for a response before the connection times out.	10000 ms
Publishing Interval	The minimum amount of time between updates.	1000 ms
Keep Alive Time	The maximum amount of time between updates..	5000 ms
Lifetime	How long the connection to the OPC UA server is preserved in the absence of updates before it is killed and recreated.	60000 ms

**Note:**

- All parameter values are shown in milliseconds.
- Entering negative values, or values below the minimum allowable, result in an error.

5. Check **Publishing Enabled** check box to receive updates from the OPC UA server. You should only uncheck this check box to temporarily disable subscriptions.
6. Click **Save**.

Database Service Providers

The database service provider allows you to connect to data sources and retrieve or update information for SQL and ODBC statements and procedures.

Database Connections

You can connect to your established data sources through either a SQL connection or an ODBC connection.

Licenses

**Note:**

You can have 100 database connections with a total of 500 statements.

Licenses are based on ODBC drivers. You also require one dedicated license to connect to SQL Server through SQL provider. Each data source requires one license, but in Demo mode, you can have up to two licenses.

Service Provider Details

Property	Value	Description
Provider	N/A	<p>If you are using a SQL connection, from the drop-down, select <code>SQL Data Provider</code>. You can use this connection method for the following data source:</p> <ul style="list-style-type: none"> • Microsoft SQL Server
Provider	N/A	<p>If you are using an ODBC connection, from the drop-down, select <code>ODBC Data Provider</code>. You can use this connection method to connect to any data source for which the ODBC driver is installed:</p> <ul style="list-style-type: none"> • Microsoft Access database file • Microsoft Excel database file • ODBC database connections (plus drivers)

Connection Details

You can enter server connection information using either the simple user input section or by using an advanced connection string.

SQL: Simple Connection

Property	Value	Description
Server	Required; User-defined up to 50 characters	Specifies the server name that you want to connect to.
Database	Required; User-defined up to 50 characters	Specifies the database name that you want to connect to on the server.
Port	Numeric	Specifies the port number of the server.
Timeout	Numeric	Specifies the amount of time allotted for the server to connect before timing out. The default time is 30 seconds.

ODBC: Simple Connection



Note:

Access restrictions: Workflow supports only system DSNs, which allow access to any user who has administrative privileges on a particular machine. User and file DSNs are not supported.

Property	Value	Description
DSN	N/A	Specifies the data source name for server drivers. To configure DSN, go to: Start menu > Control Panel > All Control Panel Items > Administrative Tools > Data Sources (ODBC) . Click the Add button, and then follow the DSN instructions.

SQL and ODBC: Advanced Connections

Property	Value	Description
Connection String	Alphanumeric	Allows you to add more advanced connection parameters,

Property	Value	Description
		such as security controls (passwords) and connection pooling.

Credentials

For your server connection login credentials, you can select either:

- SQL Server Authentication OR
- Windows Authentication

If you select **SQL Server Authentication**, you must indicate a user name and password.

Property	Value	Description
User Name	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 50 characters	Specifies the user name used to authorize connection to the server.
Password	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 50 characters	Specifies the password related to the user name to connect to the server.

ODBC Authentication

Property	Value	Description
User Name	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 50 characters	Specifies the user name used to authorize connection to the server.
Password	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 50 characters	Specifies the password related to the user name to connect to the server.

Testing the Connection

You can use the **Test Connection** button to ensure that the database connection details and credentials will work in a production environment.

Database Statements

You can use statements to either update or query a database.



Note:

You can have 100 database connections with a total of 500 statements.

Writing Statements

You use the **Designer** tab to develop statements to execute.

Property	Value	Description
Statement Type	N/A	<p>Specifies the type of statement you want to execute. Use the drop-down list to select from one of the following types:</p> <ul style="list-style-type: none"> • Query, to select data • Update, to insert, update, and delete data <div data-bbox="1105 1087 1419 1446" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note: If you choose Query (to select data) after the Update query, the query to select does not run.</p> </div>
Provider	N/A	<p>Specifies the data provider that you want to connect to. Use the drop-down list to select from one of the following providers:</p>

Property	Value	Description
		<ul style="list-style-type: none"> • SQL Data Provider • ODBC Data Provider <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: For ODBC, you cannot have more than one query. </div>
Statement	String	<p>Specifies the text that defines the statement. Use a different parameter placeholder depending on the data provider that you select:</p> <ul style="list-style-type: none"> • SQL uses @ • ODBC uses ? <p>Parameter placeholders are delimited by spaces. For example: Select * from EMPLOYEE where EMPID = @ID;</p> <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: Data transfer limit: 10 MB. </div>
Create Parameters	N/A	When you write a statement, click Create Parameters to add parameters automatically to the parameters list.
Add	N/A	Click to add a new parameter to the list. Select a field to change the properties of the parameter:

Property	Value	Description
		<ul style="list-style-type: none"> In the Name column, enter a parameter name. In the Data Type column, use the drop-down list to select the related data type.
Delete	N/A	Click to remove a parameter from the list. Select a row or rows, and then click this button.

Testing Statements

You can test the statements developed in the **Designer**. You can call the service provider method using the statement and an existing connection in the data model. You can also specify test parameter values. The results, such as statement execution time and error messages, are displayed when the test completes.



Note:

For more information on ODBC error messages, go to:

[http://technet.microsoft.com/en-us/library/aa937531\(v=sql.80\).aspx](http://technet.microsoft.com/en-us/library/aa937531(v=sql.80).aspx)

Property	Value	Description
Inputs	N/A	Specifies the parameter name and value based on the designed statements.
Connection	N/A	Click the Browse button to open the Universal Browser , and then select a server to connect to.
Test	N/A	Click the Test button to test the statements. <div data-bbox="1036 1759 1084 1806" style="float: left; margin-right: 5px;"> </div> CAUTION: Statements are executed against a specified con-

Property	Value	Description
		 nection, which may result in loss of data.
Outputs	N/A	<p>Specifies the statement data that is returned after execution. When there are multiple queries as part of the database statement, the output consists of a collection of result sets. You can select information from the results set drop-down list to display the specific results in the data grid. If you change the connection, the outputs also change. Outputs include:</p> <ul style="list-style-type: none"> • Execution Status • Execution Time • Rows Affected/Returned <p>Depending on the statement type, outputs can be either <i>Rows Affected</i> or <i>Rows Returned</i>. When the statement type is:</p> <ul style="list-style-type: none"> • <i>Query</i>, then the output specifies the result set(s), which are displayed in the table and the total number of <i>Rows Returned</i> are also displayed. • <i>Update</i>, then the output specifies the <i>Rows Affected</i>.

Create a SQL database connection

You can create a basic database connection to use for configuring SQL service provider statements and procedures.

About this task



Note:

You can have 100 database connections with a total of 500 statements.



Note:

You can have 100 database connections with a total of 500 statements.

Procedure

1. In the navigator, click **Proficy System > Proficy System > Databases > Database Connection**.
2. Click **Add Connection**.
3. In the **Name** and **Description** fields, enter a name and description for the resource.
4. In the **Displays** panel, click **Database Editor**.
5. From the **Provider** drop-down list, select **SQL Data Provider**.
6. In the **Connection Details** section, select one of the following radio buttons.
 - To fill predefined fields, click **Simple**.
 - To enter a user-defined string, click **Advanced**.
- 7.

If you selected...	Enter...
Simple	a. In the Server field, enter a server name. b. In the Database field, enter a database name. c. In the Port field, enter a port number. d. In the Timeout field, enter time in seconds.
Advanced	◦ A SQL connection string for your database connection.

8. From the **Authentication** drop-down list, select **SQL Server Authentication**.
9. In the **User Name** field, enter a name.
10. In the **Password** field, enter a password.
11. Click **Save**.

Configure a stored procedure for using the SQL service provider

After creating a database connection, you can configure stored procedure for a SQL service provider.

Procedure

1. To execute a stored procedure, open a workflow, and then insert and configure a Call Method activity.
2. From the call method activity configuration panel:
 - a. Click **Configure Method**.
 - b. Expand **IDatabaseService**, select **ExecuteProcedure**, and then click **Next**.
 - c. In the **Procedure** section, click **Browse** next to **Connection**.
The **Universal Browser** appears.
 - d. Select a database connection, and then click **OK**.
The available user-defined stored procedures are displayed in the **Procedure** drop-down list.
 - e. Select a procedure to execute, and then click **Finish**.
 - f. From the call method activity configuration panel, in the **Inputs** section, you can configure any required input parameters.



Note:

The **Connection** and **ProcedureName** fields are read only.

- g. Click **Save**.
3. **Optional:** For optional parameters, you can pass null values. To do so, perform the following procedure.
 - a. Bind the parameter to a String local variable.
 - b. Insert a Write activity.
 - c. Set the local variable to null from an Object local variable that has a null setting.
 - d. Click **Save**.
4. **Optional:** You can also execute a procedure from WPF and Silverlight forms by invoking a server method. For more information, see *Bind a return value to a DataGrid using form server methods*.

Configure a database statement for a SQL service provider

After creating a database connection, you can configure database statement for a SQL service provider.

Procedure

1. In the navigator, click **Proficy System > Proficy System > Databases > Database Statement**.
2. Click **Add Statement**.
3. In the **Name** and **Description** fields, enter a name and description for the resource.
4. In the Displays panel, click **Database Editor**.
5. Click the **Designer** tab.
6. From the **Statement Type** drop-down list, select one of the following actions for a statement.

Select...	To...
Query	select
Update	insert, update, or delete

7. From the **Provider** drop-down list, select **SQL Data Provider**.
8. In the **Statement** section, enter a valid SQL string, and then click **Create Parameters** to generate automatically any parameters used in the statement.
9. **Optional:** To add other parameters, in the data grid, click **Add**.
10. **Optional:** To configure the data grid properties:
 - In the **Name** column, select a cell, and then enter a new name.
 - In the **Data Type** column, select a cell, and then from the drop-down list, select a data type.



Note:

The parameters must match those used in the statement string.

11. To remove a parameter, in the data grid, select a row, and then click **Delete**.
12. Click **Save**.
13. To test the statement:
 - a. Click the **Test** tab.
 - b. In the **Inputs** section, in the **Parameter Value** field, enter a value for each parameter.
 - c. In the **Connection** section, click **Browse**.
The **Universal Browser** appears.
 - d. Select a database connection to run the statement against, and then click **OK**.
 - e. In the **Test** section, click the **Test** button.

**CAUTION:**

Statements are executed against a specified connection, which may result in loss of data.

The **Outputs** section displays the test results.

14. To execute a statement, open a workflow, and then insert and configure a Call Method activity.

15. From the call method activity configuration panel:

- a. Click **Configure Method**.
- b. Expand **IDatabaseService**, select **ExecuteStatement**, and then click **Next**.
- c. In the **Statement** section, click **Browse** next to the **Use Existing** radio button.
The **Universal Browser** appears.
- d. Select a statement, and then click **Finish**.
- e. On the call method activity configuration panel, in the **Connection** field, click the ellipsis button.
The **Universal Browser** appears.
- f. Select a database connection, and then click **OK**.
- g. To bind the connection to a workflow or subprocess property, click the drop-down arrow, and then select **Bind**.
- h. From the Call Method activity panel, in the **Inputs** section, you can configure any required input parameters.

**Note:**

The **Statement** field is read only.

- i. Click **Save**.

16. **Optional:** You can also execute a procedure from WPF and Silverlight forms by invoking a server method. For more information see **Bind a return value to a DataGrid using from server methods**.

Bind a return value to a DataGrid using form server methods

To display a data table in a Silverlight form, you must bind the return value using server methods.

Before you begin

Create a Silverlight form for web-based content.

About this task**Note:**

The information in this topic is not related to WPF forms unless otherwise stated.

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Silverlight Config**.
4. Click **Add**.
Your **Windows** browser appears.
5. From the <install dir>\Microsoft SDKs\Silverlight\v5.0\Libraries\Client folder, select `System.Windows.Controls.Data.dll`, and then click **Open**.
6. **Optional:** In the **Description** field, enter new information about the file, and then click **Save**.
7. Click the **Designer** tab.
8. Click the **Server Methods** tab.
9. Click **Add**.
10. In the **Properties** panel, in the **Source** field, click the ellipsis button.
The **Pick a method** dialog box appears.
11. Expand **IDatabaseService**, select **ExecuteStatement**, and then click **Next**.
12. In the **Statement** section, click **Browse** next to the **Use Existing** radio button.
The Universal Browser appears.
13. Select a statement, click **OK**, and then click **Finish**.
14. In the **Connection** field, click the ellipsis button.
The Universal Browser appears.
15. Expand **Databases > Database Connection**, select a connection, and then click **OK**.
16. From the **EventTriggers** field, click the **Add** button.
The **Event Triggers** dialog box appears.
17. Select the event that you want to trigger the method.
18. Click **Save**.
19. Click the **Project** tab.
20. Expand <YOURPROJECTNAME> > **MainPage.xaml**.
21. Double-click **MainPage.xaml.cs**.
22. Add the code in bold to this page in the location indicated. Edit column names and add additional columns as appropriate for your call.

```

namespace YOURPROJECTNAME
{
    public partial class MainPage : UserControl, IFragment
    {
        [Import]
        public IFeedbackService FeedbackService { get; set; }

        readonly ILog _logger = LogManagerEx.GetLogger("YOURPROJECTNAME");
        readonly Dictionary<Type, object> _interfaceHelpers = new Dictionary<Type, object>();

        public MainPage()
        {
            ProficyComposition.SatisfyImports(this);
            InitializeComponent();

            _interfaceHelpers[typeof(IResourceParameters)] = new FragmentMetadataByAttributes(this);
            _interfaceHelpers[typeof(IFragmentInitialization)] = new FragmentInitialization(this);

            InitializeProficyData();

            DataGrid1.AutoGenerateColumns = false;

            DataGrid1.Columns.Add(new DataGridTextColumn() { Header = "Header1", Binding = new
            Binding("[YOURCOLUMNNAME]") });

            DataGrid1.Columns.Add(new DataGridTextColumn() { Header = "Header2", Binding = new
            Binding("[YOURCOLUMNNAME2]") });

        }

        public T GetInterface<T>() where T : class
        {
            if (_interfaceHelpers.ContainsKey(typeof(T)))
            {
                var iface = _interfaceHelpers[typeof(T)] as T;

                Debug.Assert(iface != null, "_interfaceHelpers[" + typeof(T).Name + "] didn't have an object of the right
                type.");

                return iface;
            }
        }
    }
}

```

```

return this as T;
}
}
}

```

23. Click **Save**.
24. Click the **MainForm.xaml** tab.
25. Click the **Toolbox** tab.
26. Expand **Custom Controls**, and then select **DataGrid**.
 - For a WPF form, expand **Grids and Panels**.
27. Drag the DataGrid control onto `MainPage.xaml`.
28. From the **Properties** panel, go to the **ItemsSource** property, click the drop-down arrow, and then select **Bind**.

The **Binding** dialog box appears.
29. Select **Server Methods** > <YOURMETHODNAME> > **ReturnValue** > **FirstTable** > **Rows**.
 - For a WPF form, bind the **InputTable** property of the Data Grid control to the result of the server method.
30. Click **Save**.

Data Types

Data types define the format that data is stored in and displayed as for variables, specifications, and other resources.

The following table describes the simple and complex data types that are supported.



Note:

Some data types are not available for some model properties or resource variables.

Data Type	Name	Definition
Boolean (bool)	Boolean	Represents the simplest data type. Booleans can have one of two values: 0 or 1. This may be displayed as True/False, On/Off, Yes/No, and so on.
Byte (byte)	Unsigned 8-bit integer	Stores positive 8-bit values, represented as integers from 0 through 255.

Data Type	Name	Definition
Byte[] (byte[])	BLOB (binary large object)	A collection of binary data stored as a single entity in a database management system.
Data Table (DataTable)	Data Table	An on-screen display of information in a database management system, in a column format, with field names at the top.
DateTime (DateTime)	Date Time	Represents an instance in history, from year through millisecond. DateTime fields range in value from 12:00:00 midnight January 1, 0001 to 11:59:59 December 31, 9999.
Double (double)	Double Precision Float (64 bits)	<p>Values range from -1.79769313486232e308 through 1.79769313486232e308, as well as positive or negative zero (PositiveInfinity and NegativeInfinity) and not a number (NaN).</p> <div data-bbox="1024 1205 1425 1381" style="border: 1px solid orange; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Important: Doubles can maintain only 15 digits of precision.</p> </div>
GUID	GUID (globally unique identifier)	128-bit integer that can be used wherever a unique identifier is required.
Int16 (short)	Signed 16-bit Integer	Values range from -32,768 through 32,767.
Int32 (int)	Signed 32-bit Integer	Values range from -2,147,483,648 through 2,147,483,647.
Int64 (long)	Signed 64-bit Integer	Values range from -9,223,372,036,854,775,808

Data Type	Name	Definition
		through 9,223,372,036,854,775,807.
UInt16 (ushort)	Unsigned 16-bit Integer	Values range from 0 through 65535.
UInt32 (uint)	Unsigned 32-bit integer	Values range from 0 through 4,294,967,295.
SByte (SByte)	Signed 8-bit integer	Values range from -128 through 127.
Single (float)	Single Precision Float (32 bits)	<p>Values range from -3.402823e38 to 3.402823e38, as well as positive or negative zero (PositiveInfinity and NegativeInfinity) and not a number (NaN).</p> <div data-bbox="1024 940 1422 1119" style="border: 1px solid orange; border-radius: 10px; padding: 10px;"> <p> Important: Singles can maintain only 7 digits of precision.</p> </div>
Nullable Int64	Nullable 64-bit Integer	Allows a value of NULL or a signed 64-bit integer.
Nullable Double	Nullable Double Precision Float	Allows a value of NULL or a double precision float.
String (string)	String	Represents a sequential collection of Unicode characters used to represent text.
TimeSpan (timespan)	Time Interval	Contains a Start Time and a Duration (specified by a single number), which can be defined by: year, month, day, hour, minute, second, and millisecond (decimal).

Data Type	Name	Definition
Object (object)	Object	Represents undeclared variables and variables declared without a data type.

Data Type Conversions

Data type conversions happen automatically between compatible data types in Workflow.

When a specific data type is selected as the source and is bound to a different data type, if available, Workflow automatically converts the source data type into the destination data type.

Source Data Type	Converts to...	
Boolean	Boolean, String, or Object	
String	Boolean, String, Byte, SByte, Int16, Int32, Int64, UInt16, UInt32, Single, Double, GUID, DateTime, TimeSpan, or Object	
Byte	String, Byte, SByte, Int16, Int32, Int64, UInt16, UInt32, Single, Double, or Object	
Byte[]	Byte[] or Object	
Byte[]	String, Byte, SByte, Int16, Int32, Int64, UInt16, UInt32, Single, Double, or Object	
Int16	String, Byte, SByte, Int16, Int32, Int64, UInt16, UInt32, Single, Double, or Object	
Int32	String, Byte, SByte, Int16, Int32, Int64, UInt16, UInt32, Single, Double, or Object	
Int64	String, Byte, SByte, Int16, Int32, Int64, UInt16, UInt32, Single, Double, or Object	

Source Data Type	Converts to...	
UInt16		String, Byte, SByte, Int16, Int32, Int64, UInt16, UInt32, Single, Double, or Object
UInt32		String, Byte, SByte, Int16, Int32, Int64, UInt16, UInt32, Single, Double, or Object
Single		String, Byte, SByte, Int16, Int32, Int64, UInt16, UInt32, Single, Double, or Object
Double		String, Byte, SByte, Int16, Int32, Int64, UInt16, UInt32, Single, Double, or Object
GUID		String, GUID, or Object
DateTime		String, DateTime, or Object
TimeSpan		String, TimeSpan, or Object
DataTable		DataTable or Object
Object		Boolean, String, Byte, SByte, Int16, Int32, Int64, UInt16, UInt32, Single, Double, GUID, DateTime, TimeSpan, DataTable, or Object

Example

You can use a write activity in Workflow to write an int32 parameter to a string parameter. The int is converted to a string when the write occurs; for example, a 12 becomes "12".

Archives

Archives

Archives contain the project run-time data that is no longer in use. Archived files are stored on the server's file system, and can be restored as needed.

By default, the server's file system is located at C:\Program Files\Proficy\Proficy Workflow\Program\Archives. The file system allows you to protect your data.

The Archive Details area displays the archive file name and location, the creation date and time, and the size of the file. If the client/server connection is severed, the server will continue to complete its tasks. When the connection is restored, the server will then notify the client of the change.

However, if the server shuts down before it finishes the queue, all the pending archives (as well as the one in progress) are lost. They are not recovered when the server runs again.

If any archiving process encounters a failure, the archive contains the word *Failure* instead of the archiving date. In addition, more details about the error are listed in the right pane under the section called Archiving Faults.



Note:

Pending or In Process archives cannot be deleted.

Related reference

[Import/Export Display \(on page 656\)](#)

Related information

[Proficy System \(on page 561\)](#)

Create a project data archive

Procedure

1. In the navigator, click **Proficy System > Proficy System**.
2. In the **Displays** panel, click **HW/SW > Archive Management**.
3. In the **Archive Management** area, click **Add**.
4. Enter a name and description for the archive.
5. In the **Archive Content** area, select the type(s) of data you want to archive.
6. In the **Archive Time Range** area, click the arrow next to the **End time** field, and then from the calendar, select the date and time for the archive to take effect.
7. Click **Create Archive**.

The archive model is added to a queue. Only one archive can exist at a time.

Restore project data from an archive file

Procedure

1. In the navigator, click **Proficy System > Proficy System**.
2. In the **Displays** panel, click **HW/SW > Archive Management**.
3. From the **Archives** list, select the archive file you want to restore data from.
4. Click **Restore Archive**.

The status bar indicates whether the data was successfully restored.

Delete an archive file

Before you begin



Note:

When a file is deleted, it is permanently removed from the folder where it was saved.

Procedure

1. In the navigator, click **Proficy System > Proficy System**.
2. In the **Displays** panel, click **HW/SW > Archive Management**.
3. From the **Archives** list, select the archive file you want to delete.
4. Click **Delete**.

The archive file is deleted.

Archive Management Editor

The **Archive Management Editor** allows you to choose the content, as well as the time frame of the content, you want to archive.

Archive Content	Description
Resource Links	Archives resource links.
Workflow	Archives workflow tracking data.
Regulatory Compliance	Archives audit and electronic signature records.
Material	Archives closed material lots and their child sublots that are not in-use by production data.
Production	Archives closed work requests and related data.

Attribute	Value	Description
Archives List	User-defined	Displays a list of files that can be arranged by name and creation date.
Add	N/A	Click to add a new archive file.
Delete	N/A	Click to delete the selected file.
Name	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 50 characters	Specifies a file name for the selected archive file.
Description	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 255 characters	Specifies information about an archive file.
Archive Content	System-defined	Select the type of information you want to archive.
Archive Time Range	User-defined	Specifies the last date and time for which information is archived.
Create Archive	N/A	Click to archive data and move it offline.
Archive Details	System-defined	<p>Displays the archive file name, creation, date, and size.</p> <div data-bbox="1024 1276 1419 1545" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: This information is available only when you select an existing archive file. </div>
Archive Content	System-defined	<p>Displays the type of content in the archive file.</p> <div data-bbox="1024 1682 1419 1833" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: This information is available only when you se- </div>

Attribute	Value	Description
		 Select an existing archive file.
Archive Time Range	System-defined	Displays the last date and time for the information in the archive file.  Note: This information is available only when you select an existing archive file.
Restore Archive	N/A	Click to restore the data from the selected archive.

Custom Property Types

Custom Property Type Editor

The **Custom Property Type Editor** allows you to add and edit custom property values.

The following table describes the attributes specific to this resource type.

Attribute	Value	Description
Add button	N/A	Click to add a new custom property type value.
Remove button	N/A	Click to delete an existing custom property type value.  Note: You cannot delete custom property types or values that are currently in use.

Attribute	Value	Description
Add Value field	N/A	Type the value you want to add to the specified custom property type.

Related information

[Create a custom property type \(on page 630\)](#)

[Modify custom property type values \(on page 630\)](#)

Create a custom property type

Procedure

1. In the navigator, click **Proficiency System > Proficiency System**
2. Select the **Custom Property Types** folder.
3. Click **Add Custom Property Type**.
4. Enter a name and description for your custom property type.
5. Click **Save**.

Modify custom property type values

Procedure

1. In the navigator, click **Proficiency System > Proficiency System**.
2. Navigate to the **Custom Property Types** folder.
3. Select the custom property type you want to edit.
4. From the Displays panel, click **HW/SW Configuration > Custom Property Type Editor**.
5. Select the following options to edit the values associated with the custom property type:

Click...	To...
Add...	Add a new value.
Remove	Delete an existing value.
The arrows	Adjust the display order of your custom property type values. This is the order in which the values will be shown in the property editors.

Transferring Workflow

Transferring Workflow Configurations

You can transfer configurations for Workflow resources between servers or operational environments by way of either export/import or projects.

Using export/import, you can create files that contain configurations for a discrete type or category of resource, whereas projects allow you to combine configurations for differing resource types in a single package.

General Configuration Transfer Considerations

Certain considerations apply across multiple Workflow resource types when transferring configurations.

**Note:**

In addition to the information outlined here, certain considerations apply based on type of resource configuration being transferred. For more information, see **Content-Specific Transfer Considerations**.

Data Transfer Recommendations

We recommend that you transfer content, as required, in the following order:

1. OPC Server
2. Equipment Model
3. Material Model
4. Personnel Model
5. Security
6. Events Model
7. Proficy System (Computers)
8. Default Computer Configuration
9. User Forms and Displays
10. Production Model
11. Workflow Model

Version Compatibility

Beginning with Workflow 2.2, you can import content that has been exported from previous versions of the software. For workflows, you can import content exported from version 1.5 Service Pack 2 or later; other types of content must have been exported from version 2.0 or later.

Links to External Data

Certain types of Workflow resources can be associated with properties or variables, which, in turn, can be linked to data items housed in external systems. When transferring configurations for such resources, you must first ensure that the destination server is configured for connection to the external system, and that the data items are present and equivalently named on that system; otherwise, linking will break upon transfer of the resource configurations to the destination server.

For example, the originating application server is connected to an OPC server named OPC_Server1, and OPC_Server1 has three groups named Group1, Group2, and Group3. The equipment properties you are transferring are linked to OPC items in Group1 named Item1, Item2, and Item3. In order for the transfer to succeed, the destination OPC_Server1 should be exported from the source system and imported into the destination system, so that the groups and tags exist with the same ID.

The following resource types may be subject to this external linking dependency when transferring configurations:

- Equipment

For more information on bulk equipment uploads, see **Equipment Provisioning**.

- Events
- Material
- Personnel
- Production
- Property Groups

Security Settings

For resource types that are subject to Workflow security enforcement, any security settings established for a related resource instance are not included when transferring the resource configuration. Rather, you must transfer these settings separately. For more information, see the topic on **transferring security configurations**.

Third-party Software Requirements

You must have Microsoft Excel 2007 installed and activated, as well as its Primary Interop Assemblies, in order to transfer certain types of Workflow resource configurations. You must also ensure that your custom language settings (if language is different than English) are set appropriately. For more information, see:

- <http://msdn.microsoft.com/en-us/library/kh3965hw.aspx>
- www.microsoft.com/downloads/details.aspx?familyid=59DAEBAA-BED4-4282-A28C-B864D8BFA513&displaylang=en
- <http://technet.microsoft.com/en-us/library/cc179091.aspx>

If you are transferring resource configurations that were created with non-Latin characters, you can open the content file in Excel2003, but the non-Latin characters will not display properly. To display non-Latin characters, you must use Excel 2007.

Configuration Updates

When previously transferred configurations change, you can update destination systems accordingly by transferring the modified configurations again from the source system.

The manner of updating configurations on the destination system varies based on resource type. In the majority of cases, configurations on the destination system are updated to reflect their modified counterparts from the source system. For example, if you transfer the configuration for an equipment instance from one system to another, change the name of the instance on the source system, and then transfer the modified configuration for that instance to the destination system, the instance name on the destination system is changed accordingly. However, if a property is associated with the equipment instance, and you change the name of that property on the source system, the property name is not updated on the destination system when the modified equipment instance configuration is transferred. Rather, the property name is left intact on the destination system, and a new property is added to the equipment instance on the destination system. Likewise, if you delete a property from the equipment instance on the source system, the property is left intact on the destination system when the modified equipment instance configuration is transferred.

Related information

[Equipment Provisioning \(on page 658\)](#)

OPC Server Configurations

OPC servers, including all child groups and tags, can be exported and imported into different systems. The IDs will be consistent between the two systems, ensuring all data links continue to work.

Possible Issues and Solutions

OPC Server with Same Name/Prog ID and Different ID

When an OPC server exists with the same name or Prog ID, but a different ID, the import will fail. The existing server must be deleted before importing the new one. By doing so, existing workflow call methods can lose bindings.

OPC Server with Same ID on Target System

When an OPC server with the same ID exists on the target system, changes, such as new groups and tags from the import are merged in. Nothing will be deleted from the target server.

OPC Server IDs on Two Separate Systems

Two separate systems with different OPC server IDs cannot import changes directly from one system to the other.

Deleting OPC Servers with Active Subscriptions

OPC servers cannot be deleted if there are active subscriptions to the data items. As a result, conditional events tied to OPC data must be disabled to allow the OPC server to be deleted, and then re-enabled after the OPC server has been imported.



Note:

If you do not use **Project** to import OPC data, ensure OPC data is imported before importing workflows, equipment, and events.

Transferring Different Types of Configurations Together

You can use projects to combine configurations for various Workflow resource types in a single package for transfer between servers or operational environments.

A project provides the means of transferring entire applications at once. The first time you do so, the project configuration and the resource configurations in the project are imported to the destination system. As an application evolves, you can change the project configuration on the source system, and then redeploy the project to the destination system. In this case, the transfer is handled as follows:

- The pre-existing project configuration on the destination system is replaced with the modified project configuration.
- The pre-existing resource configurations on the destination system are updated to reflect their counterparts in the modified project. For more information, see configuration updates in: **General Configuration Transfer Considerations**.
- Any previously transferred resource configurations that are not included in the modified project are left intact on the destination system.

If you expect to update and redeploy projects over time, it is important to closely monitor project versions. The protocol used to track project iterations is `major.minor.build.revision`. Of these components, `major` and `minor` are user-specified, whereas `build` and `revision` are system-controlled. The build number increases by 1 each time a project is packaged, and the revision number increases by 1 with each change to project composition and each time an updated project is saved. Before redeploying an updated project, you should carefully review the contents of the project file, the information about the file's origin, and the notes about the project currently deployed on the system, as displayed in the deployment interface. These measures are especially important to ensure that you are deploying the intended version of the project.

For more information on the types of resources that you can transfer configurations for, see **Content-Specific Transfer Considerations**.

**Note:**

If you want to transfer one or more files that contain configurations for only a single type of resource, you can export/import instead.

Related reference

[Project Editing and Packaging Display Details \(on page 643\)](#)

[Project Deployment Display Details \(on page 647\)](#)

Related information

[Combine resource configurations in a project \(on page 636\)](#)

[Package a project to export its contents \(on page 637\)](#)

[Deploy a project to import its contents \(on page 638\)](#)

[Change project attributes or contents \(on page 640\)](#)

[Delete a project \(on page 642\)](#)

[General Configuration Transfer Considerations \(on page 631\)](#)

[Transferring Configurations by Resource Type \(on page 652\)](#)

Combine resource configurations in a project

The first step in transferring varying types of resource configurations together is to create a project, and then add the configurations to it.

Before you begin

To perform this task, you must be a member of a group that is assigned a key set with permission to do the following:

- Create a project
- For more information, see: **Workflow Security**

About this task

Before you begin transferring workflow configuration, so the following:

Procedure

1. In the navigator, click Proficy System > Deployment > Projects.
2. Click **Add Project**.
3. In the **Name** field, enter a unique name for the project.
4. In the **Description** field, enter clarifying information about the project.
5. Click **OK**.
6. In the **Displays** panel, select **Deployment Configuration > Project Editor**.
The **Project Editor** appears.
7. Click the **Content** tab.
8. Click the **Add Content** button.
The **Universal Browser** appears.
9. Locate and select the resource that you want to include a configuration for in the project file.



Note:

Use **Shift+click** to select contiguous items or **Ctrl+click** to select individual items. Certain content, though selectable here, cannot be exported; therefore, it is excluded from the project upon packaging. For more information, see **Content-Specific Transfer Considerations**.

10. Click **OK**.
11. Repeat steps 8-10 until all required content is added to the project.
12. Click **Save**.

Package a project to export its contents

After you have created a project and populated it with resources, you must package the project as the next step in transferring the related configurations. When you package a project, a compressed file containing the related configurations is exported to a location that you designate.

Before you begin

To perform this task, you must be a member of a group that is assigned a key set with permission to do the following:

- Package a project
- For more information, see **Workflow Security**.

Prerequisites

[Combine resource configurations in a project \(on page 636\)](#)

About this task

Before you package a project, combine resource configurations in a project.

Procedure

1. In the navigator, click **Proficy System > Deployment > Projects**.
2. In the project list, click the project that you want to package.
3. In the **Displays** panel, select **Deployment Configuration > Project Editor**.
The **Project Editor** appears.
4. In the **Project Editor**, click **Package**.
The **Package Project** dialog box appears.
5. Navigate to the location where you want to save the project file, change the default **File name**, if necessary, and then click **Save**.
 - The focus in the **Project Editor** shifts to the **Results** tab, which displays messages about the progress and outcome of the packaging process.



Note:

You can transfer the messages displayed on the Results tab for review and retention elsewhere by right-clicking within the tab, selecting Copy to Clipboard, and then pasting the copied content in the application of your choice.

- The project file is saved in .zip format to the specified location.
- The **History** area on the **Details** tab of the **Project Editor** refreshes and displays the package version, the time of packaging, and the user who performed the action.

Related reference

[Project Editing and Packaging Display Details \(on page 643\)](#)

Deploy a project to import its contents

After a project has been packaged, you must deploy the resulting project file as the final step in transferring the configurations it contains.

Before you begin

To perform this task, you must be a member of a group that is assigned a key set with permission to do the following:

- Deploy a project
- Form more information, see **Workflow Security**.

Prerequisites

[Transferring Different Types of Configurations Together \(on page 634\)](#)

[Combine resource configurations in a project \(on page 636\)](#)

[Package a project to export its contents \(on page 637\)](#)

About this task

Before you can deploy a project, transfer configuration, combine resource configurations, and package a project.



Important:

Prior to executing the deployment action as the last step of this procedure, you should carefully review the contents of the project file, the information about the file's origin, and the notes about the project currently deployed on the system (if applicable). These measures are especially important if the project has been updated over time, to ensure that you are deploying the intended version of the project.

Procedure

1. Log in to Workflow on the server machine that you want to transfer packaged resource configurations to.
2. In the navigator, click **Proficy System > Deployment > Projects**.
3. In the **Displays** panel, select **Deployment Configuration > Project Deployment**.
The project deployment interface appears.

4. Click the **Details** tab.
5. Beside the **Project File** field, click the ellipsis button.
The **Select Project Package** dialog box appears.
6. Navigate to the location where the project file is stored, select the file, and then click **Open**.
7. In the **Notes** area, review the information about the project file's origin.

**Attention:**

If the project is already deployed on the system, information about the currently deployed project is displayed. If the project file was created from a version of the project that is older than the version currently on the destination system, the Notes area displays a warning icon as well as clarifying text. Before you proceed with deployment, carefully review the configuration information on both the Details and Content tabs to ensure that the package contains the configurations that you want to transfer. For more information, see **Project Deployment Display Details**).

8. Click **Deploy** (if this is an initial deployment of the project) or **Update** (if this is a subsequent deployment).
 - The focus in the **Project Deployment** display shifts to the **Results** tab, which displays messages about the progress and outcome of the deployment process.

**Note:**

You can transfer the messages displayed on the Results tab for review and retention elsewhere by right-clicking within the tab, selecting Copy to Clipboard, and then pasting the copied content in the application of your choice.

- Resource configurations that do not pre-exist on the destination system are added there; existing configurations are updated. (For more information, see configuration updates in **General Configuration Transfer Considerations**.)

Related reference

[Project Deployment Display Details \(on page 647\)](#)

Related information

[General Configuration Transfer Considerations \(on page 631\)](#)

Change project attributes or contents

You can update a project by adding or removing Workflow resource configurations and/or modifying project attribute values.

Before you begin

To perform this task, you must be a member of a group that is assigned a key set with permission to do the following:

- Edit a project
- For more information, see **Workflow Security**.

Prerequisites

[Combine resource configurations in a project \(on page 636\)](#)

About this task

Before you change project attributes, combine resource configurations.

Procedure

1. In the navigator, click **Proficy System > Deployment > Projects**.
2. In the list of definitions, select the project that you want to update.
3. In the **Displays** panel, select **Deployment Configuration > Project Editor**.
The **Project Editor** appears.
4. Update the applicable project components:

To update...	Do the following...
Project name and/or description	<ol style="list-style-type: none"> a. Click the Details tab. b. Click the Edit link. c. In the Name and/or Description fields, enter new values for these attributes.
Project version information	<ol style="list-style-type: none"> a. Click the Details tab. b. To change the major version, enter the new value in the left Version field. c. To change the minor version, enter the new value in the right Version field.



Note:
The protocol used to track project iterations is `major.minor.build.revision-`

To update...	Do the following...
	<p> . Of these components, major and minor are user-specified, whereas build and revision are system-controlled. The build number increases by 1 each time a project is packaged, and the revision number increases by 1 with each change to project composition and each time an updated project is saved.</p>
<p>Project configuration content</p>	<p>a. Click the Content tab.</p> <p>b. To add configurations, click the Add Content button to open the Universal Browser, locate and select the resources that you want to add configurations for, and then click OK.</p> <div data-bbox="938 957 1424 1446" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note:</p> <p>Use Shift+click to select contiguous items or Ctrl+click to select individual items. Certain content, though selectable here, cannot be exported; therefore, it is excluded from the project upon packaging. For more information, see Content-Specific Transfer Considerations.</p> </div>

To update...	Do the following...
	<p>c. To delete configurations, select the corresponding grid rows, and then click Delete Content.</p> <div data-bbox="940 422 1422 644" style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note: Use <code>Shift+click</code> to select contiguous items or <code>Ctrl+click</code> to select individual items.</p> </div>

5. Click **Save**.

Delete a project

If a project is no longer relevant to your operations, you can delete it.

Before you begin

To perform this task, you must be a member of a group that is assigned a key set with permission to do the following:

- Delete a project
- For more information, see **Workflow Security**.

Prerequisites

[Combine resource configurations in a project \(on page 636\)](#)

About this task

Before you delete a project, combine resource configurations.



Note:

When you delete a project, it is permanently removed from the system. However, any resource configurations that were included in the project are not deleted.

Procedure

1. In the navigator, click **Proficy System > Deployment > Projects**.
2. In the list of definitions, click the project that you want to delete.

3. Click **Delete**.
4. Click **OK** to permanently delete the project.

Project Editing and Packaging Display Details

The **Project Editor** allows you to populate projects with differing types of resource configurations and then export the results in preparation for transfer to another system.

Common Controls

Regardless of the tab that is currently active in the **Project Editor**, three controls are always displayed.

Property	Value	Description
Save	N/A	Click to update the database with your modifications.
Cancel	N/A	Click to discard any changes and open the most recently saved version of the project.
Package	N/A	Click to initiate the process of exporting the resource configurations in the project. (For more information, see Package a project to export its contents .)

Details Tab

The **Details** tab displays summary information about the project and includes mechanisms for changing the project name, description, major version, and minor version.

Property	Value	Description
Edit	N/A	Click to display fields for modifying the project name and/or description.
Close	N/A	Click to hide the fields after displaying them for modifying the project name and/or description.

Property	Value	Description
Version	User-defined;Read Only/System-defined	<p>Specifies the current iteration of the project.</p> <div style="border: 1px solid #00a0c0; border-radius: 10px; padding: 10px; background-color: #e6f2f8;"> <p> Note: The protocol used to track project iterations is <code>major.minor.build-revision</code>. Of these components, <code>major</code> and <code>minor</code> are user-specified, whereas <code>build</code> and <code>revision</code> are system-controlled. The build number increases by 1 each time a project is packaged, and the revision number increases by 1 with each change to project composition and each time an updated project is saved.</p> </div>
History	Read Only/System-defined	Specifies the modification and packaging dates and times for the project, as well as the user who performed the actions.

Content Tab

The **Content** tab contains a grid that lists the resource configurations currently included in the project, and includes controls for adding and removing configurations from the project.

Property	Value	Description
Type	N/A	Specifies the configuration category of the resource. For more in-

Property	Value	Description
		formation, see Content-Specific Transfer Considerations .
Name	N/A	Specifies the user-defined name of the resource.
Description	N/A	Specifies clarifying information about the resource.
Add Content	N/A	Click to open the Universal Browser to select the resources that you want to include configurations for in the project. <div data-bbox="1024 793 1419 1241" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;">  Note: Certain content, though selectable here, cannot be exported; therefore, it is excluded from the project upon packaging. For more information, see Content-Specific Transfer Considerations. </div>
Delete Content	N/A	Click to remove configurations from the project that correspond to selected resource entries.

Results Tab

The **Results** tab is populated during project packaging with messages about the actions and related outcomes of the process.



Note:

The information on this tab refreshes each time a project is packaged. You can transfer the messages displayed on the Results tab for review and retention elsewhere by right-clicking within the tab, selecting **Copy to Clipboard**, and then pasting the copied content in the application of your choice.

Property	Value	Description
Timestamp	Read Only/System-defined	Specifies the date and time that the packaging action or action outcome occurred.
Type	Read Only/System-defined	<p>Specifies the classification of the packaging action or action outcome. Possible values are as follows:</p> <ul style="list-style-type: none"> • <i>Info</i>: Indicates initiation or completion of a stage in the packaging process. • <i>Progress</i>: Indicates ongoing activity in the packaging process. • <i>Warning</i>: Indicates that a condition resulted in failure to package a resource configuration within the project. • <i>Error</i>: Indicates that a condition resulted in overall packaging failure. <div data-bbox="1105 1297 1424 1791" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note: Detailed information about errors is written to the client log file (found at <installldir>\Proficiency\Logs\Proficiency-Client.log).</p> </div>

Property	Value	Description
Message	Read Only/System-defined	Specifies a description of the packaging action or action outcome.

Related information

[Package a project to export its contents \(on page 637\)](#)

[General Configuration Transfer Considerations \(on page 631\)](#)

Project Deployment Display Details

The **Project Deployment** display allows you to open a project file and deploy the resource configurations it contains to the current system. The display shows the contents of the select project file, information about the file's origin, and notes about the project currently deployed on the system (if applicable).

Project File Selection and Deployment Mechanisms

Regardless of the tab that is currently active in the **Project Deployment** display, two mechanisms are always displayed.

Property	Value	Description
Project File	N/A	Click the ellipsis button to access the Select Project Package dialog box, and then open the project file for review before deployment.
Deploy or Update	N/A	Click Deploy (if an initial deployment) or Update (if a subsequent deployment) to import the resource configurations in the file <u>after</u> confirming that the selected file has the correct version and content.

Details Tab

The **Details** tab displays information about the origin of the project file selected for deployment and, if applicable, notes about the version of the project that is currently deployed on the system. In the latter case, the implications of deploying the currently selected project file are also displayed.

Property	Value	Description
Version	Read Only	<p>Specifies the iteration of the project that the selected file was packaged from.</p> <div data-bbox="1027 663 1419 1520" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note:</p> <p>The protocol used to track project iterations is <code>major.minor.build-revision</code>. Of these components, <code>major</code> and <code>minor</code> are user-specified, whereas <code>build</code> and <code>revision</code> are system-controlled. The build number increases by 1 each time a project is packaged, and the revision number increases by 1 with each change to project composition and each time an updated project is saved.</p> </div>
Notes	Read Only	Specifies the date and time that the selected project file was packaged, the system that the project file was packaged on, and the user who packaged the project file.

Property	Value	Description
		<p>If an iteration of the project has previously been deployed on the system, the following information is also displayed:</p> <ul style="list-style-type: none"> • Iteration of the project currently on the system • Date and time that the project iteration currently on the system was packaged • System on which the project iteration currently on the system was packaged • User who packaged the project iteration currently on the system • Date and time that the project iteration currently on the system was deployed • User who deployed the project iteration currently on the system <p>If an iteration of the project has previously been deployed on the system, and it is newer than the iteration that the selected project file was packaged from, a warning icon and text alerting you to the situation is also displayed.</p>

Property	Value	Description
		 Note: Because the build number and, therefore, the iteration of a project changes with each packaging, the warning icon and corresponding text is also displayed if you open a project file on the same system that it was packaged on.

Content Tab

The **Content** tab contains a grid that displays information about the resource configurations that are included in the selected project file and will be imported upon deployment.



Note:

Resource configurations that do not pre-exist on the destination system will be added to the system; those that do pre-exist will be updated. For more information, see Configuration Updates in **General Configuration Transfer Considerations**.

Property	Value	Description
Content Type	Read Only	Specifies the resource category of the configuration. For more information, see General Configuration Transfer Considerations .
Name	Read Only	Specifies the user-defined name of the resource.
Description	Read Only	Specifies the user-defined clarifying information about the resource.

Results Tab

The **Results** tab is populated during project deployment with messages about the actions and related outcomes of the process.



Note:

The information on this tab refreshes each time a project is deployed. You can transfer the messages displayed on the Results tab for review and retention elsewhere by right-clicking within the tab, selecting Copy to Clipboard, and then pasting the copied content in the application of your choice.

Property	Value	Description
Timestamp	Read Only/System-defined	Specifies the date and time that the deployment action or action outcome occurred.
Type	Read Only/System-defined	<p>Specifies the classification of the deployment action or action outcome. Possible values are:</p> <ul style="list-style-type: none"> • <i>Info</i>: Indicates initiation or completion of a stage in the deployment process. • <i>Progress</i>: Indicates ongoing activity in the deployment process. • <i>Warning</i>: Indicates that a condition resulted in failure to deploy a resource configuration within the project. • <i>Error</i>: Indicates that a condition resulted in overall deployment failure.



Note:
Detailed information about er-

Property	Value	Description
		 rors is written to the client log file (found at <installldir>\Proficiency\Logs\Proficiency-Client.log).
Message	Read Only/System-defined	Specifies a description of the deployment action or action outcome.

Related information

[General Configuration Transfer Considerations \(on page 631\)](#)

Transferring Configurations by Resource Type

Using export/import, you can create files that contain configurations for Workflow resources of a given type or category for transfer between servers or operational environments.

You can export configurations for only one resource type or category per file. For more information on the types of resources that you can transfer configurations for, see **Content-Specific Transfer Considerations**.



Note:

If you want to combine configurations for differing Workflow resource types in a single package for transfer between servers or operational environments at the same time, use projects instead. See **Transferring Different Types of Configurations Together**.

Beginning with Workflow 2.2, you can import content that has been exported from previous versions of the software. For workflows, you can import content exported from version 1.5 Service Pack 2 or later; other types of content must have been exported from version 2.0 or later.

Related information

[Transferring Different Types of Configurations Together \(on page 634\)](#)

Export resource configurations by type

You can export files that contain configurations for Workflow resources of a given type or category to an external folder in preparation for transfer to another server or operational environment. After you have exported the configurations, you can import them, as required.

Before you begin

Before you export resource configurations, transfer workflow configurations and transfer configurations.

Prerequisites

[Transferring Configurations by Resource Type \(on page 652\)](#)

About this task



Tip:

If you plan to transfer configurations for numerous inter-dependent resources of varying type, or expect to transfer configurations multiple times, consider using projects instead. See **Transferring Different Types of Configuration Together**.

Procedure

1. In the navigator, click **Proficiency System > Proficiency System**.
2. In the **Displays** panel, click **HW/SW Configuration > Import/Export**.
3. In the **Content Folder** field, click the ellipsis button.
The **Browse for Folder** dialog box appears.
4. Locate and select the folder you want to export your file(s) to, and then click **OK**.



Tip:

You can also create a new folder from within the **Browse for Folder** dialog box. If you plan to move inter-dependent content files from one computer to another before importing the configurations in the files to the destination server, you must move all such files. To avoid moving unnecessary files, create a unique export folder for each set of inter-dependent content files, and then move all files in the folder to the staging computer.

The folder path is displayed in the field.

5. In the **Content Files** area, click **Add File**.
A new content file is added to the list.
6. In the **Name**, enter a name for the file you want to export.
7. In the **Description** field, enter information about the file you want to export.
8. From the **Content Type** list, select the resource type that you want to export configurations for.

9. In the **Content Details** area, click **Add Content**.

The **Universal Browser** appears.

10. Locate and select the resource that you want to include a configuration for in the export file.



Note:

Use `Shift+click` to select contiguous items or `Ctrl+click` to select individual items.

11. Click **OK**.

12. To include configurations for resources of a different category within the same resource, type in the export file, repeat steps 9-11 until all required resources are added to the export file.



Note:

If you want to exclude a resource configuration from the export file, in the **Content Details** area, select the listing for that resource, and then click **Delete Content**. Use `Shift+click` to select contiguous items or `Ctrl+click` to select individual items.

13. When you are finished selecting resources to export configurations for, click **Export Content**.

The file containing the configurations is exported to the designated folder.

Related information

[Transferring Different Types of Configurations Together \(on page 634\)](#)

Import resource configurations by type

You can import Workflow resource configurations contained in files that have been exported from other systems or operational environments.

Before you begin

Before you can import resource configurations, export resource configurations.

Prerequisites

[Transferring Configurations by Resource Type \(on page 652\)](#)

[Export resource configurations by type \(on page 653\)](#)

Procedure

1. In the navigator, click **Proficiency System > Proficiency System**.
2. In the **Displays** panel, click **HW/SW Configuration > Import/Export**.

3. In the **Content Folder** field, click the ellipsis button to locate the folder where the file that contains the configurations was exported to.
4. From the **Content Files** list, select the file containing the configurations that you want to import.
5. Click **Import Content**.

Resource configurations that do not pre-exist on the destination system are added there; existing configurations are updated. (For more information, see Configuration Updates in **General Configuration Transfer Considerations**.)

Related information

[General Configuration Transfer Considerations \(on page 631\)](#)

Refresh the list of resource configuration files

Refreshing the list of resource configuration files allows you to see files that were added to the currently selected location either by other Workflow users or by a process outside of Workflow, such as when files are manually moved from one server to another in preparation for transferring resource configurations between systems by way of import.

Procedure

1. In the navigator, click **Proficy System > Proficy System**.
2. In the **Displays** panel, click **HW/SW Configuration > Import/Export**.
3. In the **Content Folder** field, click the ellipsis button to locate your content files.
The **Content Files** list displays all of the files in the selected folder.
4. Click **Refresh**.
The **Content Files** list refreshes to show all files currently in the selected location.

Delete resource configuration files

If a file that contains Workflow resource configurations is no longer needed, you can delete it.

Before you begin



Note:

When a file that contains Workflow resource configurations is deleted, it is permanently removed from the folder where it was saved.

Procedure

1. In the navigator, click **Proficy System > Proficy System**.
2. In the **Displays** panel, click **HW/SW Configuration > Import/Export**.

3. In the **Content Folder** field, click the ellipsis button to locate your content files.

The **Content Files** list displays all of the files in the selected folder.

4. Select the file that you want to delete, and then click **Delete File**.

The **Delete** dialog box appears.

5. Click **Yes** to permanently delete the file, or click **No** to cancel the deletion.

Import/Export Display

The **Import/Export** display allows you create files containing configurations for a given type or category of Workflow resource for transfer between servers or operational environments.

The following describes the available options.

Attribute	Value	Description
Content Folder	User-defined	Click the ellipsis button to select a folder to export a resource configuration file to.
Content Files	User-defined	Displays a list of resource configuration files in the selected location. <div data-bbox="1024 1094 1419 1503" style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px; background-color: #e6f2ff;">  Note: This list may not reflect all resource configuration files currently in the location. For more information, see Refresh the list of resource configuration files. </div>
Add File	User-defined	Click to create a file to add resource configurations to.
Delete File	User-defined	Click to delete the selected resource configuration file.
Refresh	User-defined	Click to update the list of resource configuration files to see all files currently in the selected

Attribute	Value	Description
		location. For more information, see Refresh the list of resource configuration files .
Name	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 50 characters	Specifies a name for the resource configuration file.
Description	User-defined alphanumeric string (A-Z, a-z, 0-9) up to 255 characters	Specifies information about the resource configuration file that a user may find helpful.
Content Type	System-defined	Select the type of resources that you want to export configurations for. For more information, see Content-Specific Transfer Considerations .
Content Details	User-defined	Displays the name and description of the resources that you added configurations for in the selected file.
Add Content	N/A	Click to open the Universal Browser to select the resources that you want to export configurations for.
Delete Content	N/A	Click to exclude the configurations for the selected resources from the export process.
Export Content	N/A	Click to send the configurations for the selected resources to the specified file in the selected folder.
Import Content	N/A	Click to transfer the resource configurations in the selected file to the current system. Resource configurations that do not pre-exist on the destination system are

Attribute	Value	Description
		<p>added there; existing configurations are updated. (For more information, see Configuration Updates in General Transfer Considerations.)</p> <div data-bbox="1024 516 1417 1138" style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note: Beginning with Workflow 2.2, you can import content that has been exported from previous versions of the software. For workflows, you can import content exported from version 1.5 Service Pack 2 or later; other types of content must have been exported from version 2.0 or later.</p> </div>

Related information

[Refresh the list of resource configuration files \(on page 655\)](#)

Equipment Provisioning

Workflow provides an efficient and effective way to import large amounts of equipment data using your SQL Server.

All provisioning commands are inserted into the SOADB database table `PR_EquipmentProvisioning.EquipmentProvisioning`. When the provisioning process is executed, all commands are processed and moved to the `PR_EquipmentProvisioning.EquipmentProvisioningResults` table. The `PR_EquipmentProvisioning.EquipmentProvisioningResults` table indicates whether the command succeeded (`status=0`) or failed (`status=1`). If a command fails, the `ErrorMessage` column describes the error that was encountered.

You can start the provisioning process by directly executing the stored procedure `[PR_EquipmentProvisioning].[usp_ProvisionEquipment]` or by running the `ProvisionEquipmentModel` call method using a workflow.

Specifying the Equipment Hierarchy

For all commands, the full hierarchy must be provided for a given equipment instance. This is done using the `Parent1`, `Parent2`, ..., `Parent9` columns. `Parent1` specifies the top level of the hierarchy (that is, enterprise), `Parent2` specifies the second level (that is, site), and so on.



Note:

A model of up to 10 levels is supported.

For example, if the command is operating on an `Area` equipment instance, you would specify the area name in the **EquipmentName** column, and provide the names of the parent enterprise and site in the **Parent1** and **Parent2** columns, respectively. If the command is operating on an enterprise equipment instance, then no parents are provided since an enterprise is at the top level.

Supported Commands

Specify the desired operation using the `Command` column. `CommandId` is internal and is generated automatically when a new record is inserted.

CommandId	Command	EquipmentName	EquipmentClassName	EquipmentType	Equipme
-----------	---------	---------------	--------------------	---------------	---------

- Create equipment instance - Command 1
 - Required Fields: EquipmentName, EquipmentType (see below for valid types), Parents
 - Optional Fields: EquipmentDescription
- Delete equipment instance - Command 2
 - Required Fields: EquipmentName, Parents
- Set property to a literal value - Command 3
 - Required Fields: EquipmentName, Parents, PropertyName, PropertyValue
- Set a Historian tag link for a property - Command 4
 - Required Fields: EquipmentName, Parents, PropertyName, PropertyValue (the Historian tag name), HistorianServerName (the display name of the Historian in SOA)



Note:

The desired tag must already be synchronized to Workflow.



Important:

After executing Command 4, run the workflow described in the KB article [000036769](#) to trigger conditional events successfully. This workflow resolves a known limitation with the Equipment Provisioning feature, which allows to create historian tag links without subscribing to events. You can also manually link the historian tag to the equipment property. However, manual linking is not recommended for large data.

- Add class to instance - Command 5
 - Required Fields: EquipmentName, Parents, EquipmentClassName
- Add property to instance - Command 6
 - Required Fields: EquipmentName, PParents, PropertyName, PropertyDataType (see below for valid types), PropertyValue
 - Optional Fields: PropertyUnitOfMeasure, PropertyDescription

Valid Values for EquipmentType (nvarchar)

- Enterprise
- Site
- Area
- WorkCenter
- ProcessCell
- ProductionUnit
- ProductionLine
- StorageZone
- WorkUnit
- Unit
- WorkCell
- StorageUnit
- EquipmentModule
- ControlModule

Valid Values for PropertyDataType (int)

- Boolean: 0
- Byte: 1
- DateTime: 3
- Double: 4

- Single: 5
- GUID: 6
- SByte: 7
- Int16: 8
- Int32: 9
- Int64: 10
- String: 11
- UInt16: 12
- UInt32: 13
- CustomPropertyType: 18
- Nullable Integer: 19
- Nullable Double: 20

Customizing Timeouts

By default, the equipment provisioning process will time out if the processing time exceeds 10 minutes. For very large numbers of commands (tens of thousands), this may be insufficient time, so the provisioning does not complete. If necessary, the timeouts can be adjusted by editing the `ProficiencyServer.exe.config` file.

The following appSetting can be added inside the `<appSettings></appSettings>` elements (the value is the timeout in seconds):

```
<add key="EquipmentProvisioningTimeout" value="1200"/>
```

If a value greater than 10 minutes is used, the WCF timeout must also be adjusted higher than the default 10 minutes.

```
<add key="OperationTimeoutSeconds" value="1200"/>
```

Limitations of Property Data Links

The equipment provisioning process supports:

- Links to Historian tags only.
- Current value links only. Any advanced query configurations must be done using Workflow client.
- Viewing and testing links in Workflow client that were created with this process. If you attempt to edit the link, it does not select the data item in the data item browser by default. This is because these links are created by a different mechanism than default data items.

Reporting

Reporting

Creating reports on data is an important part of any business and Workflow provides a reporting database for you to install that is specially designed to facilitate reporting and analysis. You can use any valid reporting tool with the reporting database and you have the flexibility to design your own ad-hoc reports.

After you install the reporting database, you can synchronize it with your production database so that you can run reports on up-to-date data.

You use the **Reporting Schedule Editor** to synchronize your databases. You can synchronize your databases based on a scheduled event, or you can synchronize manually. For example, if you need to run a report a week before the databases are scheduled to synchronize, you can use the **Synchronize Now** button on the **Reporting Schedule Editor**. Be aware, however, that the **Synchronize Now** button is not available if there are unsaved changes to the **Reporting Schedule Editor**.

**Note:**

If you plan on frequent synchronization for reporting purposes, we recommend that you replicate your database and run reporting against the replicated database. This measure reduces the risk of system timeout or shutdown due to locking of tables in the production database. If, upon trying to save data after replication, you receive an error similar to "System.Data.SqlClient.SqlException (0x80131904): Length of LOB data (88064) to be replicated exceeds configured maximum 65536," use SQL statements to increase the volume of data to replicate.

For information on setting up database replication, visit the Microsoft Developer Network website and review the article, [SQL Server Replication](#).

When upgrading the application, if you are using a replicated database and have replication turned on, you must turn it off, upgrade, and then turn it on again. You must also upgrade the replicated database.

Synchronization SQL Statements

```
sp_configure 'max text repl size', 2147483647
GO
RECONFIGURE
GO
```

Reporting Database Replication

Workflow includes an option to install a reporting database. This database can be used with any valid reporting tool, and allows you to build ad-hoc reports on up-to-date data that you synchronize from your production database.

For more information on synchronizing your databases, see **Reporting**.



Note:

If you plan on frequent synchronization for reporting purposes, we recommend that you replicate your database and run reporting against the replicated database. This measure reduces the risk of system timeout or shutdown due to locking of tables in the production database. If, upon trying to save data after replication, you receive an error similar to "System.Data.SqlClient.SqlException (0x80131904): Length of LOB data (88064) to be replicated exceeds configured maximum 65536," use SQL statements to increase the volume of data to replicate.

For information on setting up database replication, visit the Microsoft Developer Network website and review the article, [SQL Server Replication](#).

When upgrading the application, if you are using a replicated database and have replication turned on, you must turn it off, upgrade, and then turn it on again. You must also upgrade the replicated database.

Synchronization SQL Statements

```
sp_configure 'max text repl size', 2147483647
GO
RECONFIGURE
GO
```

We recommend that you install the reporting database on a separate computer from the production database to prevent performance issues on the production database. Because the reporting database is installed on a separate computer, you must configure the connection between the production server and database, and the reporting server and database. You configure this connection when you install the reporting database. For more information, see **Install the reporting database**.

If, for any reason, you need to uninstall the reporting database, you are given the choice of completely removing the database from the computer or simply disconnecting the synchronization association

between the databases. If you choose to disconnect synchronization, you can still run reports on the data that remains in the database.

The reporting database utilizes a reporting schema. To understand how to use the reporting database and generate your reports, you must understand the reporting schema defined for the Reporting Database. For more information, see **Reporting Database Schema**.

**Important:**

We assume that Reporting Database users are experienced SQL users who are familiar with the SQL Server reporting tool or other reporting tools.

**Note:**

Before you start using the Reporting Database, we recommend that you deploy the sample auditing reports that are included in the installation to view the data synchronization status.

Related reference

[Reporting Database Schema \(on page 667\)](#)

Related information

[Upgrade a replicated database \(on page 169\)](#)

[Reporting \(on page 662\)](#)

Upgrade a replicated database

When upgrading the application, if you are using a replicated database and have replication turned on, you must turn it off, upgrade, and then turn it on again. You must also upgrade the replicated database.

Procedure

1. Disable any scheduled synchronization jobs for the Reporting Database.
2. Open the **Services** window to stop all services: click **Start > Control Panel > Administrative Tools > Services**.
3. In Microsoft SQL Server Management Studio, generate the create and drop scripts:

**Note:**

For use later in this procedure, you must note the location of where you saved the scripts.

- a. In the Subscriber/Distributor instance, right-click the **Replication** folder, and then click **Generate Scripts**.
The **Generate SQL Script** dialog box appears.
 - b. To generate the **Create** script, in the **Script the commands** section, select **To create or enable the components**, and then click **Generate Script**.
 - c. To generate the *Drop* script, in the **Script the commands** section, select **To drop or disable the components**, and then click **Generate Script**.
4. To disable replication, execute the *Drop* script.
 5. Verify that all objects beneath the Microsoft SQL Server Management Studio **Replication** folder have been deleted.
 6. From the installation directory where your SQL Server is hosted, go to `Program Files \Proficy\Proficy Workflow\Program`, and then run `ProficyInstaller` to upgrade the `ProficyServer` programs and database.
 7. Edit the *Create Replication* script.
 - a. In Microsoft SQL Server Management Studio, from the **File** menu, click **File > Open File**.
 - b. Navigate to the folder where you saved the scripts generated in step 3.
 - c. Select the `Create SQL` script.
 - d. Locate the section denoted by this comment, `***** End: Script to be run at Publisher *****/`, and then add the following lines after it.

**Note:**

You must change the name of the Distribution database and the user names to match your system.

```
-- Create users in the Distribution database
USE [DistributionDB];

GO

CREATE USER [QASYSTESTSQLSRV\SqlDistAgent]
    FOR LOGIN [QASYSTESTSQLSRV\SqlDistAgent]

EXEC sp_addrolemember N'db_owner',
    N'QASYSTESTSQLSRV\SqlDistAgent';

CREATE USER [QASYSTESTSQLSRV\SqlLogReaderAgent]
```

```

FOR LOGIN [QASYSTESTSQLSRV\SqlLogReaderAgent];
EXEC sp_addrolemember N'db_owner',
    N'QASYSTESTSQLSRV\SqlLogReaderAgent';
CREATE USER [QASYSTESTSQLSRV\SqlSnapshotAgent]
FOR LOGIN [QASYSTESTSQLSRV\SqlSnapshotAgent];
EXEC sp_addrolemember N'db_owner',
    N'QASYSTESTSQLSRV\SqlSnapshotAgent';

```

8. Add all of the passwords back into the file. The Create Generation process does not write the user account passwords to the file.

- a. Find all occurrences of `@job_password = null`.
- b. Replace `null` with the log on password that precedes it. If the preceding `@job_login` is also `null`, do not edit the password so that it remains set to `null`.

```

-- Before:
exec [SystemTest2v2].sys.sp_addlogreader_agent @job_login = N'QASYSTESTSQLSRV\SqlLogReaderAgent',
    @job_password = null
-- After:
exec [SystemTest2v2].sys.sp_addlogreader_agent @job_login = N'QASYSTESTSQLSRV\SqlLogReaderAgent',
    @job_password = 'proficy'
-- Do not change passwords for null job_logins
exec [SystemTest2v2].sys.sp_addqreader_agent @job_login = null, @job_password = null

```

9. Execute the **Create Replication** script.



Note:

This script will have some errors because it will try to publish database objects that were removed from the database as part of the upgrade.

10. Re-enable the replication, and then add the new database tables and objects to the publication in order to push them to the replicated database:

- a. Expand your database directory and the **Replication** folder, right-click **Local Publications**, and then select **Properties**.
- b. Select **Articles**, and then select all unchecked items to include them in the publication.

**Note:**

The subscription must be re-initialized to include the new articles. We recommend re-initializing all articles in the subscription using a full snapshot. For instructions, read the Microsoft Developer Network article, [Reinitializing a Subscription](#)

11. After replication is working against the upgraded database, install the Reporting upgrade, and point it to the replicated database.
12. Test reporting synchronization by running a manual Sync from the Workflow client.

Reporting Database Schema

The following information describes the reporting database schema.

The reporting database schema defines the tables, the fields in each table, and the relationships between fields and tables. The reporting database schema is based on dimensional modeling. Two important concepts of dimensional modeling are facts and dimensions:

Facts

Measures that you want to report and perform analysis on. Facts are typically numeric values that can be aggregated. For example, work data, such as soldering temperature or the number of processors used in a board assembly line, are facts.

Dimensions

Attributes or descriptors that define a fact. Dimensions are the entry point to facts, and are the source of all the constraints and report criteria. For example, equipment is a dimension of work data facts because it defines the location of the Segment and Work Responses.

Dimension information can be a total count of used processor parts in one particular line.

Database Schemas

The following schema names are used in the reporting database to organize tables and simplify permission granting:

dbo

Default schema for all of the dimension and fact tables.

admin

Administration-related tables, such as system parameters.

audit

Auditing tables.

etl

ETL (Extract, Transform, and Load) staging tables.

Naming Conventions

The following naming conventions are used in the reporting database schema:

Naming Convention	Description
Fact_	Prefix for fact tables.
Dim_	Prefix for dimension tables.
Bridge_	Prefix for bridge tables. Bridge tables are used to maintain the data relation. For example, Bridge_EquipmentHierarchy allows you to easily query for all subsidiaries of an equipment resource.
SK_xxx_ID	The format of all surrogate keys in the dimension tables.
xxx_Key	The format of all foreign key fields. For example, the work request location column is named "Location_Key."
xxx_ID	The format of all the source primary keys. For example, the work request ID is named "WorkRequestID."

The reporting database allows you to generate reports on the following types of data:

- Response Work Data
- Equipment Model
- Material Model
- Personnel Model
- Workflow Model

- Auditing
- Regulatory Compliance

Notes on Dimension Tables

- The date and time dimension tables are pre-populated when the reporting database is created. These dimension tables make it easy to generate reports such as "Report on the average temperature measured in the morning compared to the afternoon" and "Report the average temperature measured on each weekday".
- The dimension tables all have an IsDeleted column. Any dimension record imported into the reporting database is kept for data integrity purposes. These records are marked as deleted by setting the flag in the IsDeleted column.
- Each dimension table uses its own surrogate key as the primary key instead of the natural primary key. The surrogate key is a number that uniquely identifies each row in the dimension table, and is used to join the dimension table into the fact table.

Related reference

[Response Work Data Tables \(on page 669\)](#)

[Equipment Model Tables \(on page 673\)](#)

[Material Model Tables \(on page 675\)](#)

[Personnel Model Tables \(on page 677\)](#)

[Workflow Tables \(on page 679\)](#)

[Audit Tables \(on page 682\)](#)

[Regulatory Compliance Tables \(on page 684\)](#)

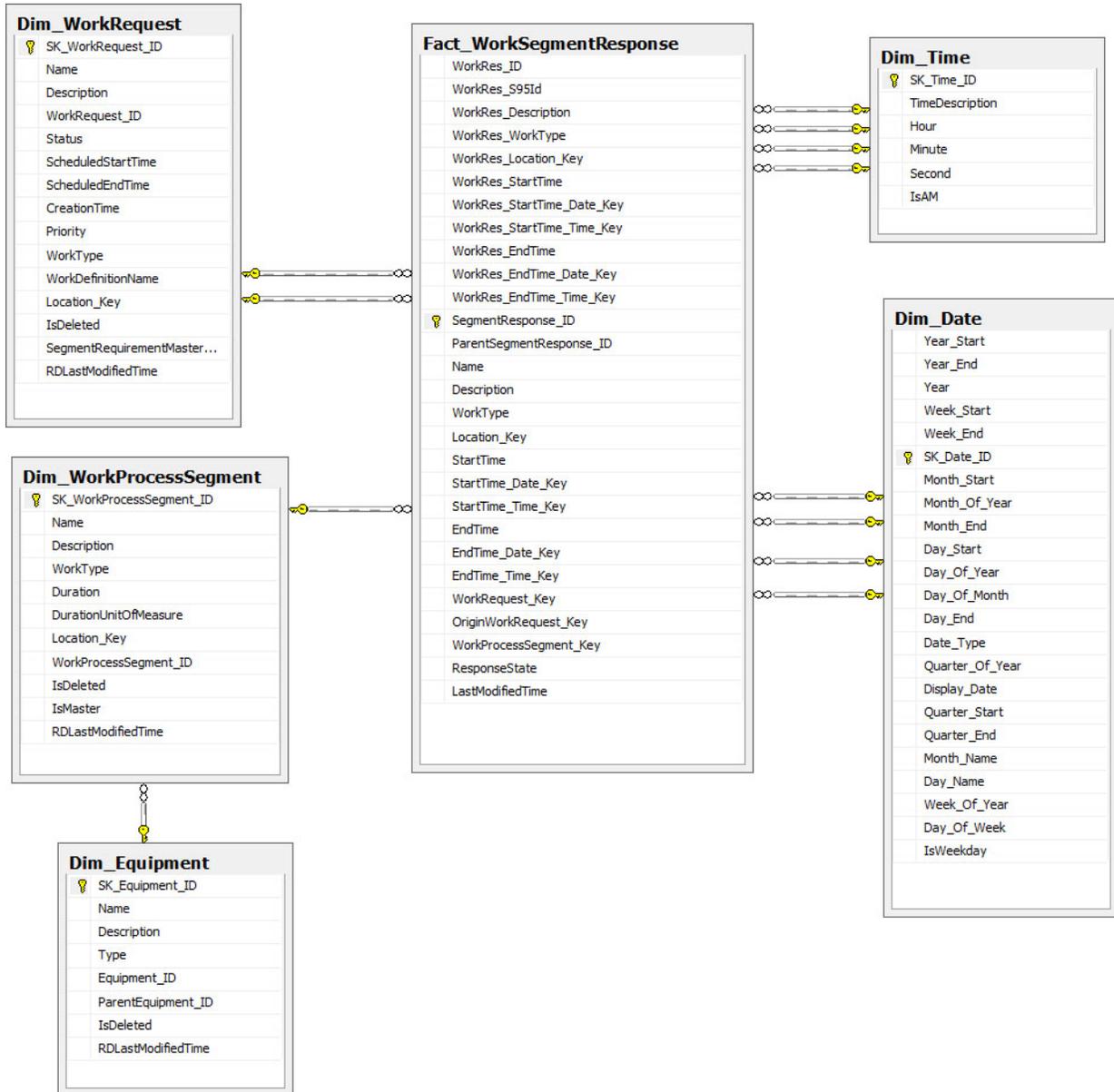
Response Work Data Tables

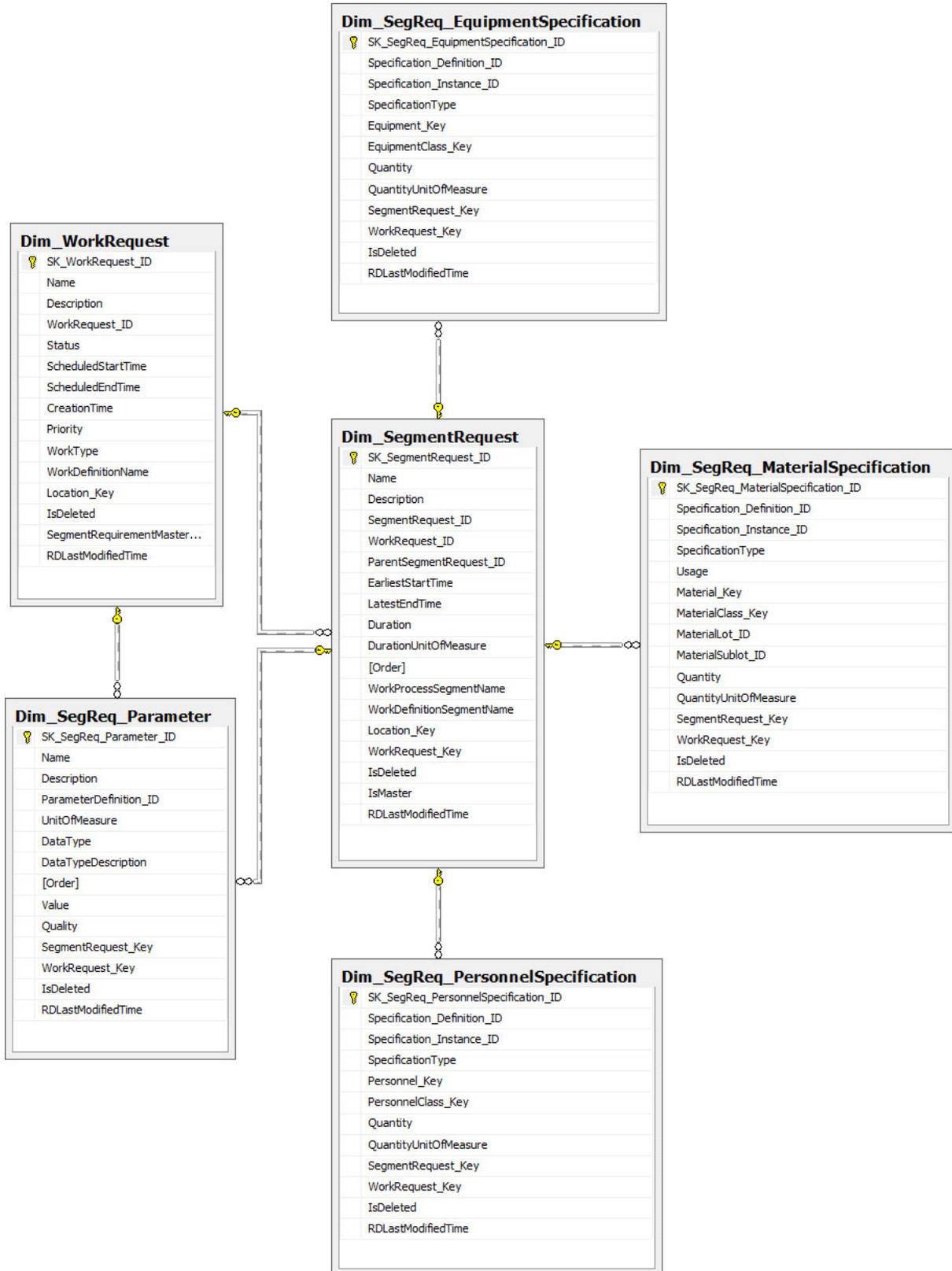
Response Work Data includes the work and segment responses. Results from the execution of a Work Request are contained in Work Responses and there may be one or more Work Responses for a Work Request. Work Responses are associated with zero or more Segment Responses. Segment Responses contain information on the individual segments of work execution for a Work Request.

The following table describes the reporting database tables and their descriptions.

Table	Description
Fact_WorkSegmentResponse	Work data associated with segment responses and work responses.
Dim_Date	Common date dimension including detailed information on each day, including the month name, day name, whether it is a weekday or not, and so on.
Dim_Equipment	Equipment definitions.
Dim_Time	Common time dimension including detailed information on time.
Dim_WorkProcessSegment	Work process segment definitions.
Dim_WorkRequest	Work request definitions.
Dim_SegmentRequest	Segment request definitions.
Dim_SegReq_Parameter	Parameter definition in segment request.
Dim_SegReq_MaterialSpecification	Material specification in segment request.
Dim_SegReq_EquipmentSpecification	Equipment specification in segment request.
Dim_SegReq_PersonnelSpecification	Personnel specification in segment request.

The following diagrams illustrate the relationship between the Response Work Data tables:





Related reference

[Reporting Database Schema \(on page 667\)](#)

Equipment Model Tables

In the Equipment model, there is no measurable data; most of the facts are relations.

The following table describes the reporting database tables and their descriptions.

Table	Description
Dim_Equipment	Equipment definitions.
Dim_EquipmentClass	Equipment class definitions.
Fact_EquipmentActual	Equipment actuals.
Fact_PropertiesOfEquipment	Equipment property configuration, including both class properties and instance properties.
Fact_ClassesOfEquipment	Equipment class and Equipment definition association.
Bridge_EquipmentHierarchy	Equipment hierarchy table.

Equipment Hierarchy Table

The Bridge_EquipmentHierarchy table is designed to simplify reports that query multiple depths of the hierarchy information. For example, you can sum the total parts used in one organization.

The bridge table contains a pathway from each parent equipment to each subsidiary below it, including itself (zero length). For example, for the following equipment hierarchy:

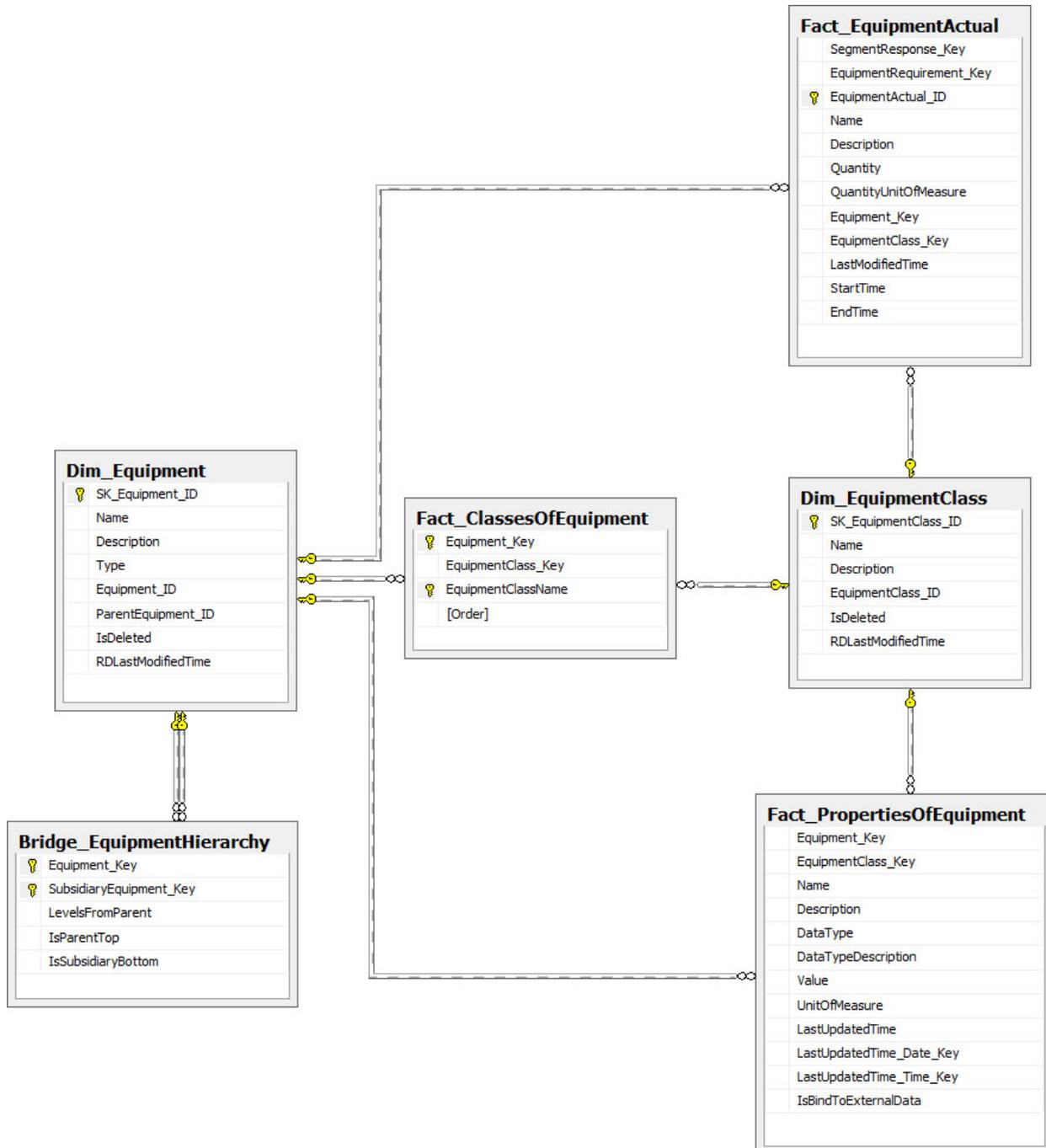
- ABC Medical Supply (1)
 - Chicago Branch (2)
 - Building 2 Production (4)
 - Detroit Branch (3)

The bridge table will look like:

Equipment	Subsidiary	Level	IsParentTop	IsSubBottom
1	1	0	Y	N
1	2	1	Y	N

Equipment	Subsidiary	Level	IsParentTop	IsSubBottom
1	4	2	Y	Y
2	4	1	N	Y
1	3	1	Y	Y
2	2	0	N	N
3	3	0	N	Y
4	4	0	N	Y

The following diagram illustrates the relationship between the Equipment model tables:



Related reference

[Reporting Database Schema \(on page 667\)](#)

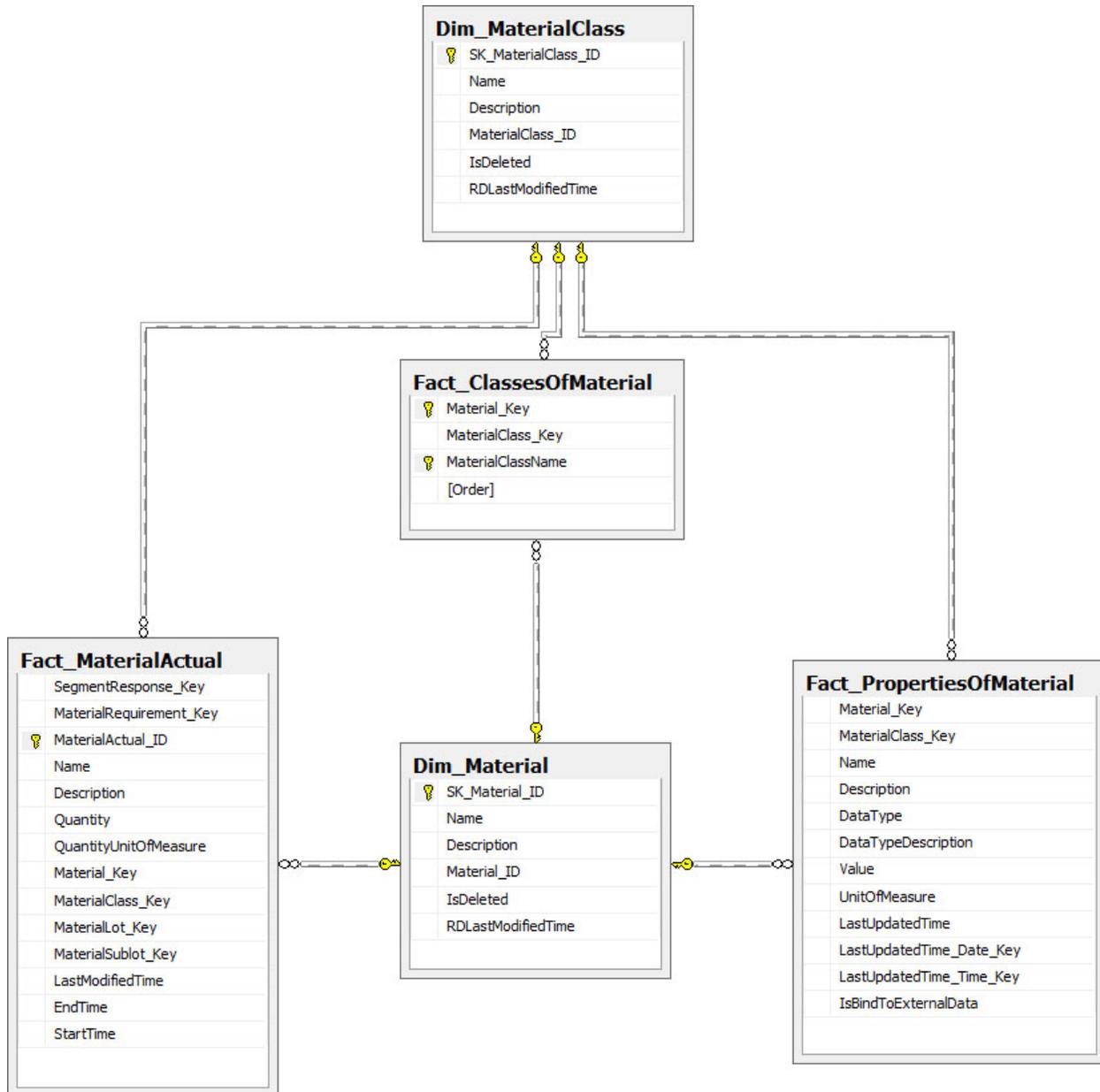
Material Model Tables

In the Material model, there is no measurable data; most of the facts are relations.

The following table describes the reporting database tables and their descriptions.

Table	Description
Dim_Material	Material definitions.
Dim_MaterialClass	Material class definitions.
Fact_MaterialActual	Material actuals.
Fact_PropertiesOfMaterial	Material property configuration, including both class properties and instance properties.
Fact_ClassesOfMaterial	Material class and Material definition association.

The following diagram illustrates the relationship between the Material model tables:



Related reference

[Reporting Database Schema \(on page 667\)](#)

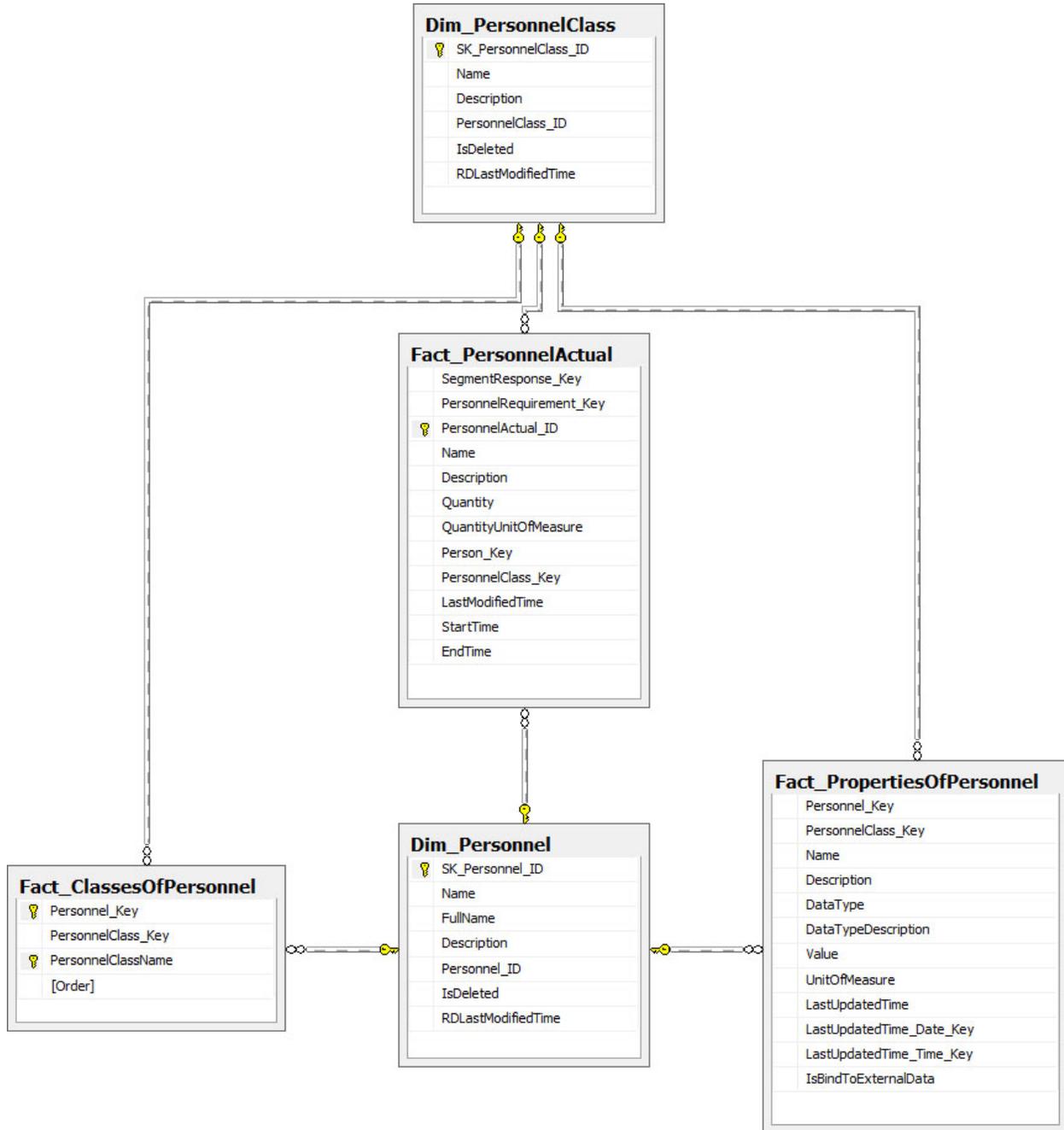
Personnel Model Tables

In the Personnel model, there is no measurable data; most of the facts are relations.

The following table describes the reporting database tables and their descriptions.

Table	Description
Dim_Personnel	Personnel definitions.
Dim_PersonnelClass	Personnel class definitions.
Fact_PersonnelActual	Personnel actuals.
Fact_PropertiesOfPersonnel	Personnel property configuration, including both class properties and instance properties.
Fact_ClassesOfPersonnel	Personnel class and Personnel definition association.

The following diagram illustrates the relationships between the Personnel model tables:



Related reference

[Reporting Database Schema \(on page 667\)](#)

Workflow Tables

The following table describes the Workflow reporting database tables and their descriptions.

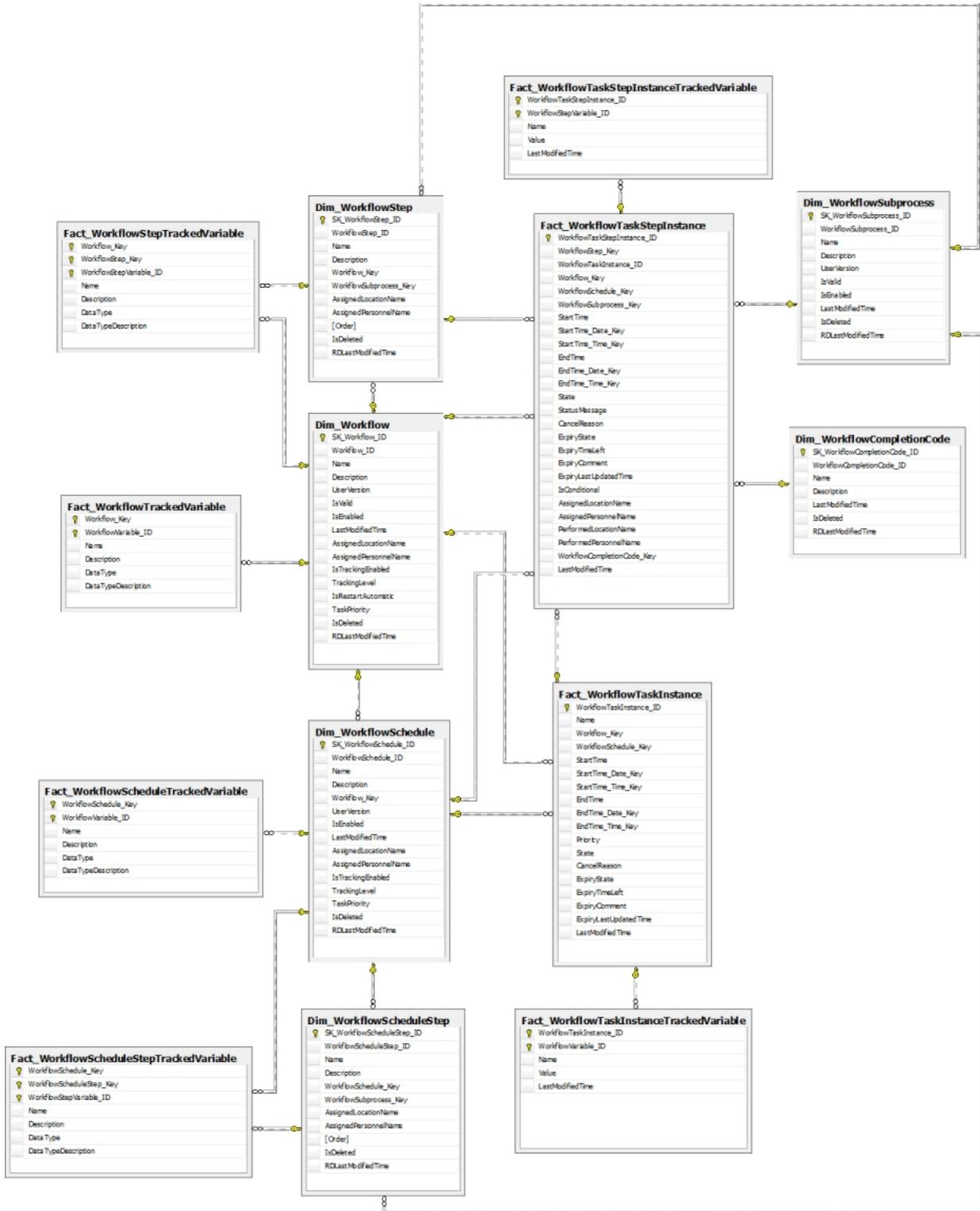
Table	Description
Dim_Workflow	Workflow definitions.
Dim_WorkflowSchedule	Schedule definitions.
Dim_WorkflowSubprocess	Global subprocess definitions.
Dim_WorkflowCompletionCode	Completion code definitions.
Dim_WorkflowCategory	Category definitions.
Fact_WorkflowsInCategory	Workflow definition and category association.
Fact_WorkflowSchedulesInCategory	Schedule definition and category association.
Fact_WorkflowSubprocessesInCategory	Global subprocess definition and category association.
Fact_WorkflowCompletionCodesInCategory	Completion code definition and category association.
Dim_WorkflowStep	Workflow step definitions.
Dim_WorkflowScheduleStep	Workflow schedule step definitions.
Fact_WorkflowTaskInstance	Workflow instance history data.
Fact_WorkflowTaskStepInstance	Workflow step instance history data.
Fact_WorkflowTrackedVariable	Workflow tracked variables.
Fact_WorkflowStepTrackedVariable	Workflow step tracked variables.
Fact_WorkflowScheduleTrackedVariable	Workflow schedule tracked variables.
Fact_WorkflowScheduleStepTrackedVariable	Workflow schedule step tracked variables.
Fact_WorkflowTaskInstanceTrackedVariable	Workflow instance tracked variables.
Fact_WorkflowTaskStepInstanceTrackedVariable	Workflow step instance tracked variables.

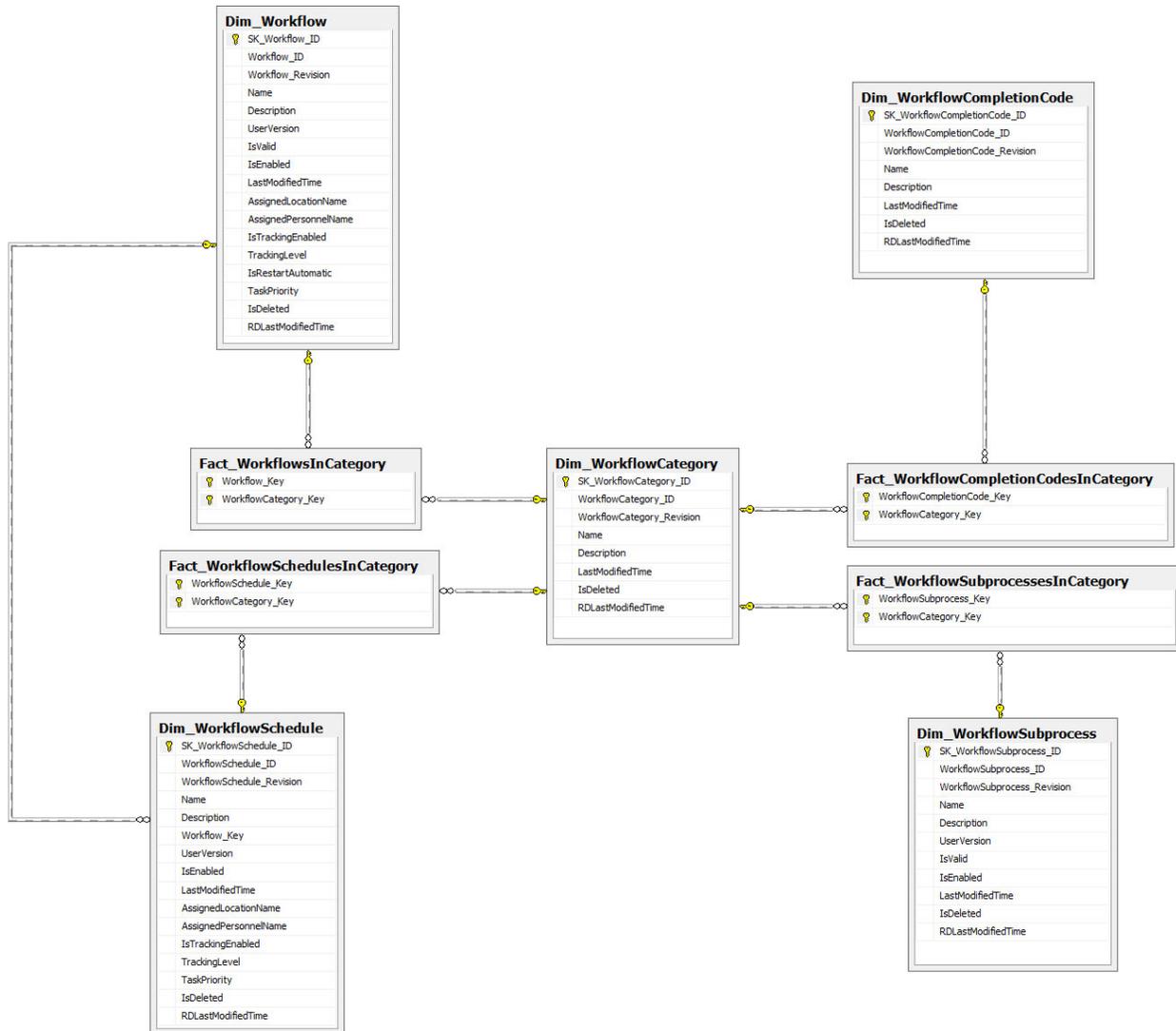
The following diagrams illustrate the relationships between the Workflow tables:



Note:

In order to report on this data, tracking configuration in the Workflow model must be set.





Audit Tables

The auditing tables are located in the audit schema.

The following table describes the reporting database tables and their descriptions.

Table	Description
ExecutionLog	Describes the top level of auditing that all records update. The status is either running, failed, or succeeded. The reporting version is also listed in the execution log.

Table	Description
PackageExecutionLog	Records each package executed. The status is running, failed, succeeded, or rolled back.
DataflowExecutionLog	Records each source table to destination/reporting table transformation.
ExecutionErrors	Includes detailed information on how many rows are fetched, deleted, updated, and inserted.

Reports can be created based on these auditing tables to show the data synchronization statistics and the current running status. This information can also be used for performance analysis and tuning.

The following diagram illustrates the relationship between the auditing tables:



Related reference

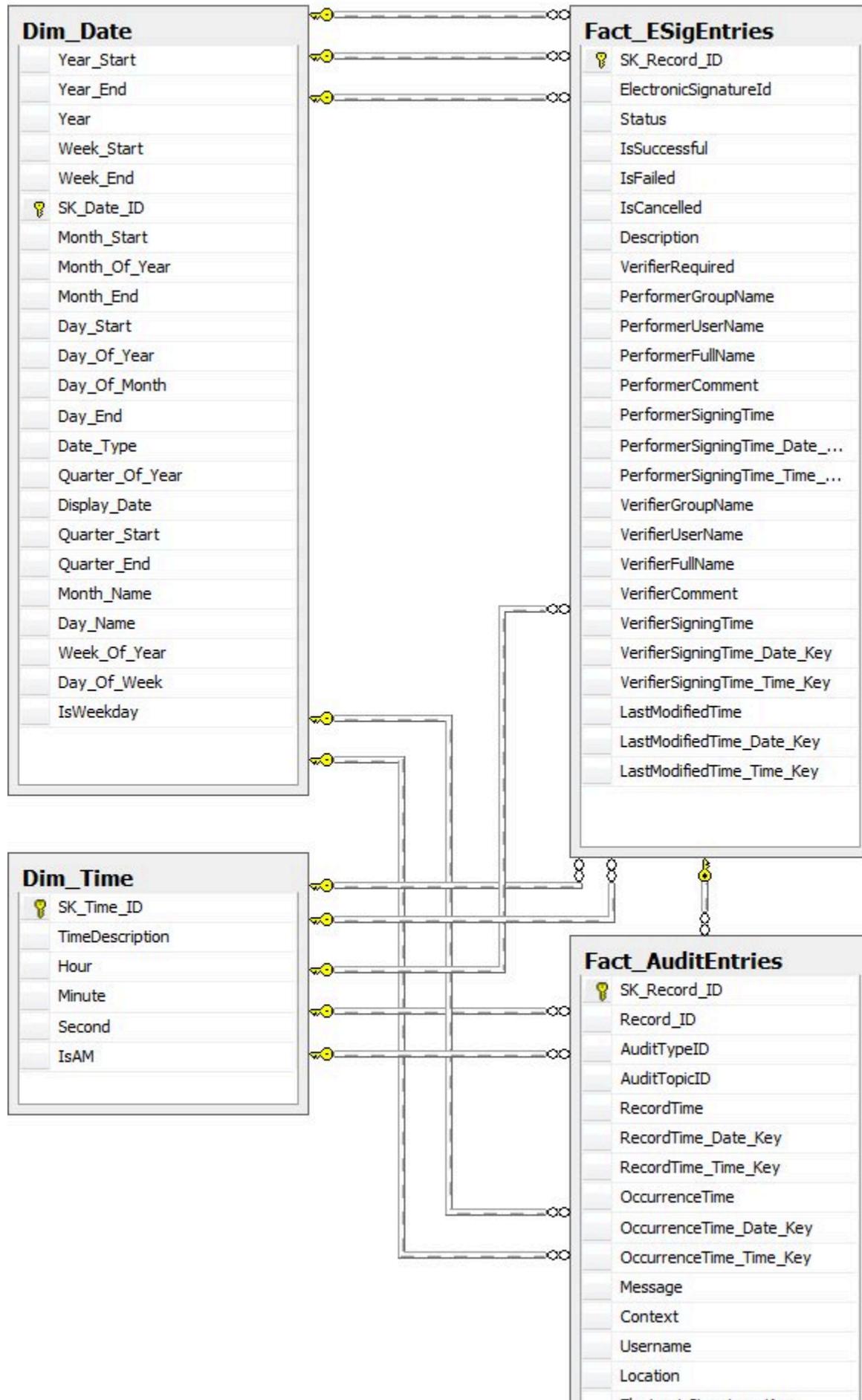
[Reporting Database Schema \(on page 667\)](#)

Regulatory Compliance Tables

The following table describes the regulatory compliance reporting database tables and their descriptions.

Table	Description
Dim_Date	Date definitions.
Dim_Time	Time definitions.
Fact_ESigEntries	Electronic signature entry information.
Fact_AuditEntries	Audit trail information.

The following diagrams illustrate the relationships between the electronic signature and audit trail tables:



Related reference

[Reporting Database Schema \(on page 667\)](#)

Default Reporting Schedule Editor

The Default Reporting Schedule editor is used to configure a connection with your reporting database, and to configure the synchronization schedule between the production and reporting databases.

Synchronizing the two databases allows you to run up-to-date reports on your data.

The following table describes the information available on the Default Reporting Schedule Editor:

Attribute	Value	Description
Server Name	User-defined	Specifies the instance name of your reporting SQL Server.
Use a Specific User Name and Password check box	User-defined	Select this check box if you configured your reporting database to use SQL Server Authentication. If this check box is cleared, Windows Authentication/Integrated Security is used.
User Name	User-defined	Specifies the user name to use for SQL Server Authentication. This field is available only when the Use a Specific User Name and Password check box is selected.
Password	User-defined	Specifies the password for SQL Server Authentication. This field is available on when the Use a Specific User Name and Password check box is selected.
Manual Synchronization: Synchronize NOW	N/A	Click to manually synchronize your databases and to run a report on current data.

Attribute	Value	Description
		<p>Manual synchronization is performed when you are between synchronization events and want to run a report on current data.</p> <p>This button is not available if there are unsaved changes to the Reporting Schedule Editor.</p>
Synchronization Events: Browse	N/A	<p>Click to open the Universal Browser where you can locate and select an event configured for synchronizing your reporting and production databases.</p> <p>If you no longer want to use a specific event, click the Delete button.</p>
Activation: Activate/Deactivate	N/A	<p>Click to activate the synchronization event you selected. Database synchronization does not occur until the schedule event is activated.</p> <p>To deactivate a schedule event, click this button when it displays Deactivate.</p>

Related reference

[Audit Trail Reports \(on page 575\)](#)

Access the Default Reporting Schedule Editor

Procedure

1. From the Navigator, click **Proficy System > Proficy System > Reporting > Default Reporting Schedule**.
2. In the **Displays** panel, click **Reporting Schedule Editor**. The **Default Reporting Schedule** editor appears.

Configure the reporting database

After initially installing the reporting database, you can change the configuration information you provided as part of that process.

About this task



Important:

- The user account used for reporting synchronization (this can be a different user than the SOA database user), must have the following SQL server roles:
 - *SQLAgentOperatorRole* for the msdb database, in order to initiate the SQL Agent job
 - *Db_owner* on the reporting database
- For performance reasons, we recommend that you install the reporting database on a different computer from Workflow. After installing and configuring the reporting database, you can synchronize it with your production database. If you plan on frequent synchronization for reporting purposes, we recommend that you replicate your database and run reporting against the replicated database. This measure reduces the risk of system timeout or shutdown due to locking of tables in the production database. If, upon trying to save data after replication, you receive an error similar to "System.Data.SqlClient.SqlException (0x80131904): Length of LOB data (88064) to be replicated exceeds configured maximum 65536," use SQL statements to increase the volume of data to replicate.



Note:

For information on setting up database replication, visit the Microsoft Developer Network website and review the article, [SQL Server Replication](#).



When upgrading the application, if you are using a replicated database and have replication turned on, you must turn it off, upgrade, and then turn it on again. You must also upgrade the replicated database.

Procedure

1. From the navigator, click **Proficy System > Proficy System > Reporting > Default Reporting Schedule**.
2. In the **Displays** panel, click **Reporting Schedule Editor**. The **Default Reporting Schedule** editor appears.
3. In the **Server name** field, enter the instance name of your reporting SQL Server.
4. If you configured your reporting database to use Windows authentication, clear the **Use a Specific User Name and Password** check box.
5. If you configured your reporting database to use SQL Server authentication, select the **Use a Specific User Name and Password** check box, and then enter the SQL Server user name and password.
6. Click **Save**.

Synchronization SQL Statements

```
sp_configure 'max text repl size', 2147483647  
  
GO  
  
RECONFIGURE  
  
GO
```

Related information

[Upgrade a replicated database \(on page 169\)](#)

Manually synchronize databases

Before you begin

You must configure your reporting database before you can synchronize it with your production database.



Note:

You cannot synchronize your databases manually if there are unsaved changes to the Reporting Schedule Editor.

Procedure

1. From the navigator, click **Proficy System > Proficy System > Reporting > Default Reporting Schedule**.
2. In the Show Compatible panel, click **Reporting Schedule Editor**. The Default Reporting Schedule editor appears in the Workspace.
3. In the **Manual Synchronization** area, click **Synchronize NOW**.

If manual synchronization starts, the Feedback Zone displays a status message that manual synchronization has started. If manual synchronization fails, the Feedback Zone displays an error message.



Note:

The Feedback Zone does not indicate whether manual synchronization is successful. You can use the reporting database auditing tables to determine that status of a data synchronization.

Synchronize databases using scheduled events

Synchronizing databases allows you to run reports on up-to-date data.

Before you begin

You must configure your reporting database and at least one event before you can schedule the data synchronization.



Note:

If you plan on frequent synchronization for reporting purposes, we recommend that you replicate your database and run reporting against the replicated database. This measure reduces the risk of system timeout or shutdown due to locking of tables in the production database. If, upon trying to save data after replication, you receive an error similar to "System.Data.SqlClient.SqlException (0x80131904): Length of LOB data (88064) to be replicated exceeds configured maximum 65536," use SQL statements to increase the volume of data to replicate.



Note:

For information on setting up database replication, visit the Microsoft Developer Network website and review the article, [SQL Server Replication](#).



When upgrading the application, if you are using a replicated database and have replication turned on, you must turn it off, upgrade, and then turn it on again. You must also upgrade the replicated database.

Procedure

1. From the navigator, click **Proficy System > Proficy System > Reporting > Default Reporting Schedule**.
2. In the **Displays** panel, click **Reporting Schedule Editor**. The **Default Reporting Schedule** editor appears.
3. To select the event you want to use for database synchronization, in the **Synchronization Events** area, click **Browse**.
The **Universal Browser** appears.
4. Click **OK**, and then click **Save**.

Synchronization SQL Statements

```
sp_configure 'max text repl size', 2147483647
GO
RECONFIGURE
GO
```

Activate and deactivate a synchronization event

Before you begin

You must selected and save a synchronization event before you can activate or deactivate it.

Procedure

1. From the Navigator, click **Proficy System > Proficy System > Reporting > Default Reporting Schedule**.
2. In the Show Compatible panel, click **Reporting Schedule Editor**. The Default Reporting Schedule editor appears in the Workspace.
3. Click **Activate**.
The Activation Status displays as activated.
4. To deactivate a synchronization event, click **Deactivate**.
The Activation Status displays as deactivated.

Reporting Datamart Tutorial

This tutorial provides a demonstration of the following:

- How to configure Reporting Services
- Verify the source of Reporting Services reports
- Deploy the reports
- Verify report deployment
- View the reports with Report Manager
- Create Workflow client displays for reports with Workflow

This tutorial uses samples from the Reporting Datamart. The samples consist of a reporting project with report definition files for a series of workflow reports.

This tutorial requires a valid Microsoft SQL Server database synchronized with the Workflow client production database, Microsoft Business Intelligence Studio (or Visual Studio), a valid Microsoft Reporting Services setup to deploy samples, and a valid Workflow installation.

Related reference

[Audit Trail Reports \(on page 575\)](#)

Install the Microsoft Reporting Services Samples

Before you begin

A reporting database must be configured before installation.

Procedure

1. Create a local folder to deploy the reports from.

**Note:**

For the tutorial, create the folder C:\PlatformDMReports.

2. Copy files from the server folder to the local folder.

**Note:**

Look for the report files in the folder: C:\Program Files\Proficy\Proficy Platform Reporting \SampleReports\Platform DMReports.

3. Review the [Post-Installation Checks for Reporting \(on page 697\)](#) after the installation completes.

Configure the Microsoft Reporting Services Samples

The following information guides you through the process of configuring Microsoft reporting services samples.

Procedure

1. Open the applicable tool based on the version of SQL Server that you are working with.

For this version...	Open this tool...
SQL Server 2012	<i>SQL Server Data Tools</i>
SQL Server 2008	<i>SQL Server Business Intelligence Development Studio</i>

2. Select **File > Open > Project Solution**.

The **Open Project** dialog box appears.

3. Navigate to the local folder where the sample report files were installed and select `PlatformDMReports.rptproj`.
4. Click **Open**.
5. Verify the content of the Microsoft SQL Server Reporting Services project.



Note:

For more information, see **Microsoft Reporting Service Samples Files** for a list of the sample report files.

6. In the `Shared Data Sources` folder, double-click `SOADM.rds`.

The **Shared Data Source** dialog box appears.

7. Click **Edit**.

The **Connection Properties** dialog box appears.

8. Complete the following fields on the Connection Properties dialog box.
 - a. In the **Server name** field, enter the server name of the Reporting Database server or leave the default name of `localhost`.
 - b. Under **Log on to the server**, select the appropriate authentication method.
 - c. In the **Select or enter a database name** field, under **Connect to a database**, set the connection to **ReportDatamartDB**.

9. Click **Test Connection**.



Note:

If the connection is not successful, repeat steps 2-8.

Related reference

[Microsoft Reporting Service Samples Files \(on page 697\)](#)

Verify the Microsoft Reporting Services Samples reports

Before you begin

You must configure the Microsoft Reporting Service Samples first.

Procedure

1. In SQL Server Business Intelligence Development Studio, in Solution Explorer, double-click `WorkflowDefinitions.rdl`.
The report opens in design mode.
2. Click the **Data** tab.
3. Click the exclamation point.
The Define Query Parameters dialog box appears.
4. Type in a parameter.



Note:

The % symbol is a wildcard for SQL Server.

5. Click **OK**.
6. Verify that the returned data is coming from your datamart. The values from your reporting database should appear.
7. Click **Preview**.
A parameters list appears.
8. Enter the appropriate parameters.
9. Click **View Report**.
A report preview appears.

Deploy the samples to a Microsoft Reporting Services Server

The following information guides you through the process of deploying Microsoft reporting services samples.

Procedure

1. Open the *Microsoft SQL Server Reporting Services Configuration* tool.
The **Reporting Services Configuration Manager - Configure Report Server** window appears.
2. Select the MicrosoftSQL Server Reporting Services Machine name and SQL Server.
3. Click **Report Server Virtual Directory**.



Note:

Reports and data source configuration files are deployed to this directory.

4. Record the name of the Report Server Virtual Directory for future reference.
5. Select **Report Manager Virtual Directory**.
6. Record the name of the directory; the URL can be used to view sample reports in an external browser such as Internet Explorer.

Verify the deployment of the reports

The following information guides you through the process of verifying report deployment.

Procedure

1. Open *Microsoft SQL Server Management Studio*.
The **Connect to Server** dialog box appears.
2. From the **Server Type** list, select **Reporting Services**.
3. From the **Server Name** list, select the name of the Microsoft SQL Server Reporting Services server.
4. From the **Authentication** list, select **Windows Authentication**.
5. Click **Connect**.

In the Object Explorer of *Microsoft SQL Server Management Studio*, two folders are created. One folder is for the SSAS Datamart source, and the other folder contains the reports.

View the sample reports using Report Manager

Procedure

1. Click **Start**, point to **Programs**, and then select **Internet Explorer**.
2. Enter the Report Manager URL.



Note:

For a computer on the Microsoft SQL Server Reporting Services node, enter `http://localhost/SQLReports`.

The reporting services home page appears with the two folders created for data sources and reports.

3. Click the **Reports** link.



Note:

Only Reports are visible.

4. Record the URL.

Create a Proficy client Display with the report

Procedure

1. Open Workflow client.
2. In the **Models** list, click **Global Displays**.
3. In the **Folders** list, click **Displays**.
4. Click **Add Display**.
5. In the **Name** field, enter a name for the display; for example, "WorkDataBySegment".
6. In the **Description** field, enter a description for the display.
7. In the **Technology** field, select **Web page**.
8. In the **Url** field, enter the URL for the sample report.
9. Click **OK**.
10. Repeat steps 3-9 for each report.

A list of reports is generated and can be viewed in the Global Viewer in the Workflow client.

Post-Installation Checks for Reporting

After installing the Proficy Client Reporting Datamart Samples, perform the following checks to determine if the installation was successful.

Open the Microsoft Server SQL Management Studio - Database Engine and look for the following:

- A database with a set of dimension and fact tables in the Tables folder of the reporting database.
- The initial Extract, Transform, Load (ETL) job, SyncPlatform, has run manually in SQL Server Agent with no errors. Search in the Jobs folder of SQL Server Agent in the Object Explorer.

Microsoft Reporting Service Samples Files

After installing the datamart reports, the following files are copied to your local folder:

File Name	Description	
SODDM.rds		Report data source file.
PlatformDMReports.rptproj		Report project file.
WorkflowDefinitions.rdl		Report definition file for a workflow report.
WorkflowSubProcessDefinitions.rdl		Report definition file for a workflow report.
WorkflowStatisticsDetails.rdl		Report definition file for a workflow report.
WorkflowOperatorStatistics.rdl		Report definition file for a workflow report.
WorkDataBySegment.rdl		Report definition file for a workflow report.

Related reference

[Audit Trail Reports \(on page 575\)](#)

Chapter 9. Forms and User Displays

Forms Introduction

Forms and User Displays Authoring Guide

Forms and user displays are features that are used throughout Workflow. Forms and user display resources are created in the Global Displays model.

Forms and user displays can be created using the Designer, or they can be uploaded from an existing .NET assembly.



Note:

- Forms and displays that were created in Visual Studio cannot be edited in the Designer.
- Because forms and user displays are distinct entities, you can create resources of these separate types with the same name.

Regardless of how a display is developed, you must first add a global displays resource using the navigator. For more information, see **Add a Global Displays resource**. After you add the display resource, you can then either select the .NET assembly and class name on the Overview tab of the Display Editor to work with an existing display, or select the Designer tab and develop a new display.

You can also duplicate displays and groups. For more information, see **Display and Group Duplication**.



Note:

Both WPF and Silverlight forms can be displayed in the Global Viewer.

Forms

Forms are displays that are used as activities in workflows, providing a user interface for task steps. A form can display data or allow you to interact with a running process, or both.

Workflow provides the ability to use basic forms included with the application, create and develop forms from scratch using the Designer, or open and use forms that were initially created in Visual Studio.

For information on using the form files included with Workflow, or for creating your own forms in Visual Studio, see **Form Templates** and **Form Development in Visual Studio**, respectively.

User Displays

Workflow allows you to add displays for your own use and to group those displays. For example, if you want to view a web page from your company Intranet within the workspace, you can create a display using the URL of the web page. This allows you to open the web page without opening a browser session outside of the application.

Form and Display Groups

You can create form and display groups to arrange your resources into meaningful categories. Groups can also contain other groups.



Note:

- You cannot add an existing form or display to a group. You can only create a form or display from within a group.
- If you delete a group, all of its nested groups and their forms and/or displays are also deleted.
- If you transfer the configuration for a form or display that is part of a group, you must also transfer the configuration for the entire group to maintain the required linking.

Form Design Environment

The Designer is a graphical design application that allows you to easily create forms and user displays and publish them to the application server where other authors can access and use them.

The **Designer** includes different types of interface controls including buttons, text boxes and blocks, check boxes, grids, panels, list and combo boxes, and much more. All of the controls have properties for configuring their look, feel, and behavior.

User displays created through the **Designer** work with the Navigator context just as the other displays do. For more information on displays, see **Display Panel**. The most commonly used user displays are reports that can be viewed in the workspace, and are not intended to be run in a workflow.

The forms you create in the **Designer** can be used to monitor real-time data and can also be used to pass new values to a workflow, which can then update the appropriate data source. The **Designer** uses Microsoft Visual Studio projects as its own project format, allowing your applications to be extended using all of the power of .NET.

**Note:**

The **Designer** supports the Visual Basic and C# programming languages.

As you work on your display, you can save it to your local machine or save it to the server.

**Important:**

You must save the display to the server before you can navigate away from the **Designer**. For example, if you want to preview a display in the Global Viewer, you must save it to the server. Be aware, however, that saving the display to the server makes it available to other authors to edit, use, or view.

If you want to preview a display without navigating away from the **Designer**, open the Preview panel, or build and run the project within the **Designer**.

**Note:**

For security reasons, you must log into Workflow to view a test build of your form or user display.

The **Designer** can connect to any of the Workflow data sources that can be accessed through a workflow; for example, Historian and equipment properties.

The **Designer** tab is comprised of the following components:

Component	Description
Toolbar	The toolbar provides the ability to perform actions within the Designer . For more information, see Designer Toolbar .
Panels	The panels allow you to organize your working space for designing and developing forms and user displays. For more information, see Display Panel .

In the **Designer**, layout is managed by top-level containers, often canvases, panels, or grids. The type of container you choose determines the style of the layout for your screen. All control editing must be done in a container.

Regardless of the type of container you choose for your layout, the same basic rule applies: an object's position is always determined by its container. In addition, a number of containers also handle the size of their child objects.

The development of a document typically begins with the selection of a container as a root element. While there are many different types of containers to choose from, WPF applications often use a canvas or a grid as the root object. Other containers are available, such as dock panel, stack panel, and uniform grid.

Designer Toolbar

The button toolbar provides the ability to perform actions within the **Designer**.

The following table describes the toolbar buttons.

Icon	Command	Description
	Build	<p>Click to build the current project.</p> <p>This button is available only in the WPF Designer.</p>
	Build and Run	<p>Click to build and run the current project.</p> <div data-bbox="1024 947 1419 1257" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: For security reasons, you must log into Workflow to view a test build of your form or user display.</p> </div> <p>This button is available only in the WPF Designer.</p>
	Save Project	<p>Click to save the current project locally.</p> <p>This button is available only in the WPF Designer.</p>
	Next item waiting to be placed	<p>Displays the icon of the selected item in the Control Toolbox. It will be the next item to be placed on the canvas.</p>

Icon	Command	Description
		This button is available in the WPF Designer and on the Layout tab of the Silverlight Designer.
	Zoom Percentage	<p>Click to select a specific zoom percentage to apply to the canvas.</p> <p>This button is available in the WPF Designer and on the Layout tab of the Silverlight Designer.</p>
	Zoom In	<p>Click to zoom in on the canvas.</p> <p>This button is available in the WPF Designer and on the Layout tab of the Silverlight Designer.</p>
	Zoom Out	<p>Click to zoom out from the canvas.</p> <p>This button is available in the WPF Designer and on the Layout tab of the Silverlight Designer.</p>
	Cut	<p>Click to cut the selected item from the canvas.</p> <p>This button is available in the WPF Designer and on the Home tab of the Silverlight Designer.</p>
	Copy	<p>Click to copy the selected item from the canvas.</p> <p>This button is available in the WPF Designer and on the Home tab of the Silverlight Designer.</p>
	Paste	Click to paste the previously cut or copied item into the canvas.

Icon	Command	Description
		This button is available in the WPF Designer and on the Home tab of the Silverlight Designer.
	Delete	<p>Click to delete the selected item from the canvas.</p> <p>This button is available in the WPF Designer and on the Home tab of the Silverlight Designer.</p>
	Undo	<p>Click to undo the previous action.</p> <p>This button is available in the WPF Designer and on the Home tab of the Silverlight Designer.</p>
	Redo	<p>Click to redo the previous action that was just undone.</p> <p>This button is available in the WPF Designer and on the Home tab of the Silverlight Designer.</p>
	Font Family	<p>Click to change the font of the currently selected item.</p> <p>This button is available only in the Silverlight Designer, on the Home tab.</p>
	Font Size	<p>Click to change the font size of the currently selected item.</p> <p>This button is available only in the Silverlight Designer, on the Home tab.</p>
	Background: Color	Click to change the background color of the currently selected item.

Icon	Command	Description
		This button is available only in the Silverlight Designer, on the Home tab.
	Foreground: Color	<p>Click to change the foreground color of the currently selected item.</p> <p>This button is available only in the Silverlight Designer, on the Home tab.</p>
	Bold	<p>Click to bold the font of the currently selected item.</p> <p>This button is available only in the Silverlight Designer, on the Home tab.</p>
	Italic	<p>Click to italicize the font for the selected item on the canvas.</p> <p>This button is available only in the Silverlight Designer, on the Home tab.</p>
	Fill: Color	<p>Click to change the fill color for the selected Shape item in the canvas.</p> <p>This button is available only in the Silverlight Designer, on the Application Settings tab.</p>
	Stroke: Color	<p>Click to change the border color for the selected item in the canvas.</p> <p>This button is available only in the Silverlight Designer, on the Application Settings tab.</p>

Icon	Command	Description
	Stroke Thickness	<p>Click to change the border stroke width for the selected shape item in the canvas.</p> <p>This button is available only in the Silverlight Designer, on the Application Settings tab.</p>
	Decimal Places	<p>Click to change the number of decimal places that are used to display numbers.</p> <p>This button is available only in the Silverlight Designer, on the Application Settings tab.</p>
	View Source	<p>Click to view the source code behind the current form or user display.</p> <p>This button is available in the WPF Designer and on the Home tab of the Silverlight Designer.</p>
	View Panels	<p>Click to select the panels you want to make available in the canvas.</p> <p>This button is available in the WPF Designer and on the Home tab of the Silverlight Designer.</p>
	Send to Back	<p>Click to send the selected item to the back of the canvas. This button is used when items are stacked or overlapped.</p> <p>This button is available in the WPF Designer and on the Layout tab of the Silverlight Designer.</p>

Icon	Command	Description
	Move Backward	<p>Click to move the selected item backwards on the canvas. This button is used when items are stacked or overlapped.</p> <p>This button is available in the WPF Designer and on the Layout tab of the Silverlight Designer.</p>
	Move Forward	<p>Click to move the selected item forward on the canvas. This button is used when items are stacked or overlapped.</p> <p>This button is available in the WPF Designer and on the Layout tab of the Silverlight Designer.</p>
	Bring to Front	<p>Click to send the selected item to the front of the canvas. This button is used when items are stacked or overlapped.</p> <p>This button is available in the WPF Designer and on the Layout tab of the Silverlight Designer.</p>
	Alignment	<p>Click to align multiple controls together. Controls are aligned by top, bottom, width, and height. The last control selected is used as the basis for alignment. This button is available only when two or more controls are selected using Ctrl + click.</p> <p>Alignment for the Silverlight Designer is controlled by four sepa-</p>

Icon	Command	Description
		<p>rate buttons: Alignleft, Alignright, Aligntop, and Alignbottom.</p> <p>The WPF Designer Alignment buttons allow manipulation of controls by aligning right, left, top and bottom. The WPF Designer alignment button provides a list of actions that can be performed when the button is clicked.</p>
	Resize	<p>Click to enable the resize handles on the selected item. Click and drag the resize handles horizontally, vertically, or diagonally.</p> <p>This button is available in the WPF Designer and on the Layout tab of the Silverlight Designer.</p>
	Rotate	<p>Click to enable the rotate handles on the selected item. The handles turn from blue to yellow.</p> <p>This button is available in the WPF Designer and on the Layout tab of the Silverlight Designer.</p>
	Rotate Left	<p>Click to rotate the selected item to the left (counterclockwise) by 90 degrees. The handles turn from blue to yellow.</p> <p>This button is available only in the Silverlight Designer, on the Layout tab.</p>
	Rotate Right	<p>Click to rotate the selected item to the right (clockwise) by 90</p>

Icon	Command	Description
		<p>degrees. The handles turn from blue to yellow.</p> <p>This button is available only in the Silverlight Designer, on the Layout tab.</p>
	Skew	<p>Click to enable the skew handles on the selected item. The handles turn from blue to red.</p> <p>This button is available in the WPF Designer and on the Layout tab of the Silverlight Designer.</p>
	Flip Horizontal	<p>Click to flip the selected control horizontally.</p> <p>This button is available in the WPF Designer and on the Layout tab of the Silverlight Designer.</p>
	Flip Vertical	<p>Click to flip the selected control vertically.</p> <p>This button is available in the WPF Designer and on the Layout tab of the Silverlight Designer.</p>
	Visibility	<p>Click to make a control show or hide selected controls in the canvas.</p> <p>This button is available only in the Silverlight Designer, on the Layout tab.</p>
	Select All	Click to select all the objects in the active layer.

Icon	Command	Description
		This button is available only in the Silverlight Designer, on the Layout tab.
	Show Grid	<p>Click to show or hide the grid lines.</p> <p>This button is available in the WPF Designer and on the Application Settings tab of the Silverlight Designer.</p>
	Snap to Grid	<p>Click to make the item conform to ruler lines, which aid in positioning it.</p> <p>This button is available only in the Silverlight Designer, on the Application Settings tab.</p>
	Search	<p>Click the arrow next to Search to search for a control in the current canvas.</p> <p>This button is available in the WPF Designer and on the Home tab of the Silverlight Designer.</p>

Designer Panels

Designer Panels

The **Designer** workspace is comprised of a number of panels that are used for designing your forms and user displays. Each panel performs a different function and may or may not be required for your design purposes.

The **Designer** allows you to define a panel layout for your purposes. The layout you define remains in place until you change it or reset to the default layout.

The **Designer** provides a set of default panels that appear in a default layout. The default layout consists of three panel shells with default panels contained within each shell. Every panel in the **Designer** can be resized.

The following table describes the available panels:

Panel	Description
Data Items*	Allows you to specify and configure the data items to be used in your project.
Default Properties	Allows you to configure the default properties for all forms or user display projects.
Document Tree*	Displays a tree view of the objects and controls included in the XAML file that was last selected.
Form Events*	<p>Allows you to define the events to be triggered from this form and that will be exposed to a workflow using this form.</p> <div data-bbox="821 989 1422 1119" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: This panel is available for forms only. </div>
Parameters*	<p>Allows you to configure input and output parameters for the selected control.</p> <div data-bbox="821 1255 1422 1386" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: This panel is available for forms only. </div>
Preview	Allows you to preview the display.
Project*	Displays a tree view of the project files and folders.
Properties*	Allows you to configure the properties for each control object , as well as for each parameter, data item, event, and so on.
Test*	Allows you to enter the input parameter values and/or select data item values for the display you want to test. After building and running, you can then view the results in this panel. The Test pan-

Panel	Description
	<p>el represents how the display will appear, and if you are developing a form, how it will behave in the Workflow Task List.</p> <div data-bbox="821 422 1422 558" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: This panel is available for forms only. </div>
Toolbox*	Allows you to select the control objects you want to add to the project.
Server Methods	Allows you to select and configure service provider methods to use in your display.
Client Methods	Allows you to select and configure Client methods to use in your display.

* Indicates a default panel that is available for both forms and user displays, except where indicated.

Related reference

[Navigate URI \(on page 1099\)](#)

Designer: Project Panel

The Project panel in the Designer allows you to view the project structure and provides the ability to create new XAML and library (DLL) files, open existing files, and add new folders and/or classes to your project. You can also choose to build or build and run your project. These tasks can all be performed from the right-click menu.

The Designer provides a template for the project, which contains all of the references and code required for a project. Forms and user displays that you create with the Designer are compatible with Visual Studio 2008 and Visual Studio 2008 Express. You can create a form or user display in the Designer and then edit it in Visual Studio.

The Project panel displays a tree view of the project files for the selected resource. When you create a new project, it is set with default folders and files. The following table describes the files and folders located on the Project panel:

Folder/File	Description	
Properties	Displays the project properties, such as general information about the .NET assembly, the GUID of the project, and version information for the .NET assembly.	
References	Contains a list of reference files that describe the .NET assembly.	
Resources	Contains two subfolders—Images and Video—and a file named DataItems.dita, by default. The Images and Video folders are used to store images and video clips that you can use in your project.	

You can configure your project to delete all of the local files when you exit the Designer. When you right-click in the Project panel, the Delete local copy of this project when Editor is closed option determines whether or not to delete the local files. If you want to edit or debug your project files in Visual Studio, clear this option.

Modify project properties

Procedure

1. In the navigator, click **Global Displays**.
2. In the **Folders** list, select the resource type you want to work with.



Tip:

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click the **Project** tab.
5. Expand the **Properties** folder.

- Right-click the **Assemblyinfo.vb** file, and then click **Open**.

The file opens in a new tab in the designer.

- Make the required changes to the project properties.
- Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Add project reference files

Before you begin

Procedure

- In the navigator, click **Global Displays**.
- In the **Folders** list, select the resource type you want to work with.



Tip:

As more lists appear, select the appropriate resource until you find the one you want to work with.

- In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
- Click the **Project** tab.
- Right-click the **References** folder, and then click **Add reference**.
The Add Reference dialog box appears.
- Locate and select the reference files you want to add to the form project, and then click **Open**.
- Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Add project resources

Before you begin**Procedure**

1. In the navigator, click **Global Displays**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click the **Project** tab.
5. Expand the **Resources** folder.
6. To add image files, right-click the **Images** folder, and then click **Add New Image**.
The **Add Image** dialog box appears.
7. To add video files, right-click the **Video** folder, and then click **Add New Video**.
The **Add Video** dialog box appears.
8. Locate and select the image or video file(s) you want to add, and then click **Open**.
9. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Delete the local copy of the project**Before you begin****Procedure**

1. In the navigator, click **Global Displays**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.

The editor appears in the workspace, displaying the resource information.

4. Click the **Project** tab.

5. Right-click the form name (that is, the top level), and then select **Delete local copy of this project when Editor is closed**.

When you navigate away from the current form project, the local copy is deleted.

Extending with Code Behind

The Designer supports Visual Basic and C# for code behind.



Note:

Object names are important in the Designer because only named objects can be used in data binding.

The Designer automatically creates a name for control objects when they are added to the design surface. Objects that are on the design surface that are containers are often not named by the Designer. You can name an unnamed object or change the name of an existing object in the Properties panel.

If you want to access an object on the design surface from your .NET code behind, you must reference it using its name. For example, if you have a button named 'Button1' on the design surface, you could access it programmatically with the following code:

```
Button1.Background = Brushes.Red;
```

Designer: Document Tree

The Document Tree panel contains the document tree that corresponds to the XAML document you are currently editing. The document tree is a hierarchal visual representation of all the elements in your workspace.

The purpose of the document tree is to:

- Provide you with a visual reference of your document's structure.
- Indicate how a control relates to other controls and sibling controls.
- Enables you to choose layers or controls within the document, in order to add to, edit, or delete these layers or controls.
- Allows you to add and move controls between layers in a drag and drop fashion.

Locate Controls

The document tree displays a hierarchical view of the controls in the current document; that is, controls that are added to another control as child controls are indented below the parent. The document tree also displays the icon of each control in your document, and when you select a control in the Designer, it is highlighted in the document tree. When you select the control in the document tree, it is also selected in the Designer; that is, it is surrounded by resize handles and anchors.

Active Layer

The Designer works in layers, maintaining an active layer at all times. You can choose to edit, add, remove, and reposition controls on the active layer only. In the document tree, the active layer is indicated by the



The following controls can be set as the active layer:

- Canvas
- Data Grid
- Dock panel
- Grid
- Stack Panel
- Uniform Grid
- Viewport 3D

In addition to the controls above, any visible object derived from content control, such as a button, can also be set as the active layer.

The following cannot be set as the active layer:

- Any object that cannot have child controls, such as an Image control.
- Any class derived from Content Control that is collapsed, such as a Combo Box Item.
- Any control whose visibility is hidden or collapsed.

Hide Layers

You can hide inactive layers by toggling the visibility of the control in the document tree or in the Designer. Hiding a layer simply modifies its visibility while you are editing the active layer.

**Tip:**

Make sure the layer is visible again before publishing your project to the server.

You can also hide individual controls within the active layer. That is, the layer itself is visible, but the controls within it are not.

Organize Content with Drag and Drop

Within the document tree you can drag and drop one control onto another. This allows cross layer assignment of controls to new parents. It also allows some properties, such as tool tips and headers, to have framework elements applied as content. Controls can also be dragged from the toolbox and dropped onto new parent controls in the document tree.

Manipulate Controls in the Document Tree

The following table describes the actions you can perform on the controls in the document tree, including the layer object(s). These actions can all be performed from the right-click context menu, while some are also available from the toolbar.

Action	Description
Locate	Indicates the location of the selected control in the Designer with a flashing orange border.
Delete	Deletes the selected control from the designer and document tree. You cannot paste a deleted control back into the Designer. Use the Undo button on the toolbar to undo the delete action.
Cut	Cuts the selected control from the Designer and document tree. If required, you can then paste the control onto the current active layer or onto another layer after making it the active layer.
Copy	Copies the selected control. If required, you can then paste the control onto the current active layer or onto another layer after making it the active layer.
Paste	Pastes a cut or copied control onto the active layer.
Paste Special	Opens the Applying <control (control name)> to Target dialog box, where you can select the properties to apply to the control. The available options depend on the control you are working with.

Action	Description
Bring to Front	Brings the selected control to the top when two or more objects are stacked.
Move Forward	<p>Moves the selected control forward by one increment when two or more controls are stacked.</p> <div data-bbox="821 495 1422 674" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;">  Note: If only two controls are stacked, this action behaves the same as Bring to Front. </div>
Move Backward	<p>Moves the selected control backward by one increment when two or more controls are stacked.</p> <div data-bbox="821 810 1422 989" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;">  Note: If only two controls are stacked, this action behaves the same as Send to Back. </div>
Send to Back	Sends the selected control to the bottom when two or more controls are stacked.
Fit to Cell	Enlarges the selected control to fit into the entire cell.
Grid	Allows you to add rows and columns, remove additional rows and columns, and set column width and row height to the top-level grid control.
Visibility	<p>Allows you to collapse or hide the selected control or layer. If a control is already hidden or collapsed, you can choose to make it visible.</p> <div data-bbox="821 1535 1422 1866" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;">  Note: Collapsing and hiding controls differ in that hiding the control reserves the space it occupies in the layout, whereas collapsing does not reserve the space. When the project is built, the space occupied by a hidden control is rendered as whitespace. </div>

Action	Description
	 instead of the control. Since space is not reserved for a collapsed control, no white-space is rendered when the project is built.
Rotate	Allows you to rotate the selected control by 90 degree increments, or to set a specific value using a slider.
Add Event Handler	<p>Opens the Event Triggers dialog box where you can choose an event to add an event handler to.</p> <p>When you select the event, the MainForm.xaml.vb panel appears where you can insert the code for the action you want to have occur when the event is triggered.</p>

Related reference

[Designer Panels \(on page 710\)](#)

Add controls to the document tree

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. In the **Folders** list, select the resource type you want to work with.
3. Click **Display Editor > Designer > Document Tree**.
4. Click the **Toolbox** tab.
5. Expand the appropriate control list, and then click and drag the control you want to add onto the document tree.

A plus sign appears below the mouse pointer when you move it over the control(s) in the document tree.
6. Release the mouse button to drop the control.
7. The Applying <control (controlname)> to Target dialog box appears.
8. Select the option you want to use to apply to the control, and then click **OK**.
9. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Locate a control

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. In the **Folders** list, select the resource type you want to work with.
3. Click **Display Editor > Designer > Document Tree**.
4. Right-click the control you want to locate, and then click **Locate**.
The control is highlighted in the Designer with a flashing orange border.

Change the active layer

By default, the top-level active layer is a grid control.

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. In the **Folders** list, select the resource type you want to work with.
3. Click **Display Editor > Designer > Document Tree**.
4. If required, add a control that can be set as the active layer by clicking **Toolbox** and expanding the group you want to work with.
- 5.
6. Click and drag the control to the desired position in the Designer.



Note:

To apply the properties of a control to another control, click and drag the control to another control in the document tree. The Applying <control (control name)> to Target dialog box appears where you can select the option you want to use to apply to the control.

7. Right-click the control in either the Designer or the document tree, and then select **Edit Layer**.

- Or -

Double-click the control in the document tree.

The control displays in the designer with an orange border, indicating that it is the active layer.

8. Add controls to the active layer, as required.

When you make another control the active layer, you cannot select the controls that were added in other active layers because they are now part of an inactive layer. In order to change these controls, you must make that layer active again.

9. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Use Paste Special

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. In the **Folders** list, select the resource type you want to work with.
3. Click **Display Editor > Designer > Document Tree**.
4. Right-click the control you want to copy, and then select **Copy**.
5. Right-click in the designer, and then select **Paste Special**.
The Applying <control (control name)> to Target dialog box appears.
6. Select the option you want to use to apply to the control, and then click **OK**.
7. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Move stacked controls

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. In the **Folders** list, select the resource type you want to work with.
3. Click **Display Editor > Designer > Document Tree**.
4. Right-click the control in the document tree, and then select the appropriate action to move the selected control within the stack..
5. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Fit a control to the grid cell

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. In the **Folders** list, select the resource type you want to work with.
3. Click **Display Editor > Designer > Document Tree**.
4. Right-click the control in the document tree, and then click **Fit to Cell**.
The control fills the entire grid cell.
5. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Control the visibility of layers and controls

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. In the **Folders** list, select the resource type you want to work with.
3. Click **Display Editor > Designer > Document Tree**.
4. To hide a control or layer, right-click it in either the document tree or the Designer, point to **Visibility**, and then click **Hidden**.

The control disappears from the Designer, but is still visible in the document tree.



Tip:

To display the control again, right-click it in the document tree, point to **Visibility**, and then click **Visible**.

5. To collapse a control or layer, right-click it in either the document tree or the Designer, point to **Visibility**, and then click **Collapsed**.

The control disappears from the Designer, but is still visible in the document tree.

**Tip:**

To display the control again, right-click it in the document tree, point to **Visibility**, and then click **Visible**.

6. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Rotate a control

Before you begin

**Note:**

You cannot rotate a layer.

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. In the **Folders** list, select the resource type you want to work with.
3. Click **Display Editor > Designer > Document Tree**.
4. Right-click the control in the document tree, point to Rotate, and then select the degree to rotate, or use the Custom slider to specify the degree.
5. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Add an event handler to a Form activity

Event handlers are activity sequences that are triggered by a specific event. They allow a workflow to run logic when the events occur while the form is active.

Before you begin



Note:

External Events, a Create Event activity, or a form event must be defined in order to add an event handler.

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow. For more information, see **Add a resource to the workflow model**.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Drag and drop a **Local Subprocess** into the workflow and double-click it, or double-click an existing subprocess.
6. Select the Form activity and then, on the activity configuration panel, click **Add Event Handler**.
The **Select Event** dialog box appears.
7. Click the **Activities, Subprocess, or Workflow** tab, select an event, and then click **OK**.
8. Click **OK**.

Designer: Control Toolbox

The Designer supports native .NET 4.5 controls and containers, third party .NET 4.5 controls, and has many of its own .NET controls available. The Designer does not support Silverlight controls or 2D shapes or drawing capabilities.



Note:

Beginning with Workflow 2.5, .NET 4.5 is required to develop display or form assemblies. Custom display or form assemblies that target earlier versions of .NET will continue to function as before. However, saving changes made to such custom assemblies requires that they be upgraded, which is accomplished in different ways, based on the version of .NET:

- For an assembly that pre-dates .NET 4.0, you must upgrade the assembly when you open it. Failure to do so results in an error and the inability to save the edited assembly.
- For an assembly that targets .NET 4.0, the assembly is upgraded automatically when opened, and changes to such assemblies can be saved as before.

Add a control

About this task



CAUTION:

If you delete or rename a control, parameter, or server method that is used as the source of a binding in a form, the binding itself is not removed from the XAML file. You can leave the binding code in the XAML, but if you create another control, parameter, or server method with the same name as the deleted object, or rename an existing object to make the object name valid again, then the orphaned binding may become valid as well, causing unexpected results.

It is recommended that you manually delete the binding code from the XAML file to prevent this issue from occurring.

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. Expand the control group you want to work with.
6. Click and drag the control to the desired position in the **Designer**.



Note:

To apply the properties of a control to another control, click and drag the control to another control in the document tree. The Applying **<control (control name)> to Target** dialog box appears where you can select the option you want to use to apply to the control.

7. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Configure control properties

Before you begin**Note:**

Some properties are available only when a control is added as a child to another control. For example, the **Dock** property is available only when a control is added as a child to a **Dock Panel** control.

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. Select the control you want to configure.
6. Click the **Properties** tab.
7. Configure the properties, as required.

**Note:**

To view the full list of properties, select the **Advanced** check box.

8. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Set up control layout

This information applies to both the WPF and Silverlight Designers. In the WPF Designer, some .NET Winforms controls (and controls written for other technologies) have the same names, but have different properties, methods, and events exposed than their WPF counterparts.

Before you begin

Adding controls with a deliberate design in mind is the recommended approach. However, if multiple controls already exist in the designer, and they must be moved into a layout control, follow the steps below. This procedure uses the grid control with other existing controls as an example.

Procedure

1. Add a layout control, such as a grid.

**Note:**

Some controls allow only a single child to be added. Add a control that allows multiple children (such as Grid, Canvas, Dock Panel, Stack Panel).

2. In the designer, select the controls you want to add to the layout control, and then right-click and select **Cut**.
3. In the Document Tree, select the layout control added in step 1, and then right-click and select **Edit Layer**.
4. In the designer, right-click the layout control, and then select **Paste**.

**Note:**

If the controls are displayed in the Document Tree but not in the designer, then in the Document Tree, select each control and set its Margin property to 0 to have them appear in the designer.

5. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Batch Controls

Batch controls cause multiple events to occur.

Batch Generic Button



The Batch Generic Button control causes multiple events to occur each time the button is pressed.

The following table describes the properties for this control.

Property	Description
AllowDrop	Select this check box to indicate that the control can accept data that is dragged onto it.

Property	Description
Background	Expand this field to select the background color of the selected control.
BatchExecution	Specifies the controls to deploy.
BorderBrush	Expand this field to select the border color of the selected control.
BorderThickness	Specifies the border thickness of a control, in pixels.
ClickMode	<p>Specifies when the Click event occurs. Selections are:</p> <ul style="list-style-type: none"> • Hover • Press • Release
ClipToBounds	Select this check box to clip the content of this control (or content coming from the child elements of this control) to fit into the size of the containing control.
Column	Specifies the column within the grid where the control is located.
ColumnSpan	Specifies the total number of columns that child content spans within a grid.
CommandParameter	Specifies the parameter to pass to the Command property.
CommandTarget	Specifies the element on which to raise the specified command.
Content	Specifies the content displayed on the control.
ContentStringFormat	Specifies a composite string that determines how to format the Content property if it is displayed as a string.
CornerRadius	Specifies the degree to which the corners of a control are rounded.

Property	Description
Effect	<p>Click the arrow to expand the property, select the effect you want to apply to the control, and then configure the specific values for the effect. Valid values to select are:</p> <ul style="list-style-type: none"> • Blur <ul style="list-style-type: none"> ◦ Kernel Type ◦ Radius • Drop Shadow <ul style="list-style-type: none"> ◦ Color ◦ Direction ◦ Opactiy ◦ Shadow Depth
FlowDirection	<p>Specifies the direction that text and other user interface (UI) elements flow within any parent control. Selections are:</p> <ul style="list-style-type: none"> • Left to Right • Right to Left
Focusable	<p>Select this check box to indicate that the control can receive focus.</p>
FocusBorderThickness	<p>Specifies the thickness of the focus ring on the outside of the button.</p>
FocusBrush	<p>Expand this field to select the color that displays around the control's border when the control has focus.</p>
FocusVisualStyle	<p>Specifies a property that enables customization of appearance, effects, or other style characteristics that apply to this control when it has focus.</p>
FontFamily	<p>Specifies the font family for the control.</p>
FontSize	<p>Specifies the font size for the control.</p>
FontStretch	<p>Specifies the degree to which a font is condensed or expanded on the screen.</p>

Property	Description
FontStyle	Specifies the font style for the control.
FontWeight	Specifies the weight or thickness of the specified font.
ForceCursor	Select this check box to indicate that the user interface (UI) renders the cursor as declared by the Cursor property.
Foreground	Expand this field to select the foreground color.
GlareBrush	Expand this field to specify the glare effect for the control.
Height	Specifies the height of the control, in pixels.
HorizontalAlignment	<p>Specifies the horizontal alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
HorizontalContentAlignment	<p>Specifies the horizontal alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
InnerBorderBrush	Expand this field to select the color for the inner border.
InnerCornerRadius	Specifies the degree to which the corners of the inner border are rounded.
IsCancel	Select this check box to indicate that a Button or Repeat Button acts as a Cancel button. Press the Esc key to select the actual Cancel button.

Property	Description
IsDefault	Select this check box to indicate that a Button or Repeat Button is the default Submit button. Press the Enter key to select the actual Submit button.
IsEnabled	Select this check box to indicate that this control is enabled in the user interface (UI).
IsHitTestVisible	Select this check box to indicate that the control can possibly be returned as a hit test result from some portion of its rendered content.
IsManipulationEnabled	Select this check box to indicate that the manipulation events for the control are enabled.
IsSharedSizeScope	Select this check box to indicate that controls are sharing size information.
IsTabStop	Select this check box to indicate that a control is included in tab navigation.
Localization	Click the ellipsis button to open the Localization Configuration dialog box where you can configure the property to localize, the language to use for localization, and the text to localize.
Margin	Specifies the outer margin of a control.
MaxHeight	Specifies the maximum height of a control.
MaxWidth	Specifies the maximum width of a control.
MinHeight	Specifies the minimum height of a control.
MinWidth	Specifies the minimum width of a control.
Name	Specifies the identifying name of the control. The name provides a reference so that code behind, such as event handler code, can refer to a markup element during processing by a XAML processor.
Opacity	Specifies the opacity factor applied to the entire control. Valid values are 0 through 1, with up to 15 decimal places.

Property	Description
OpacityMask	Expand this field to select a color for the opacity mask, which is applied to any alpha-channel masking for the rendered content of this control.
OverridesDefaultStyle	Select this check box to indicate that this control incorporates style properties from theme styles.
Padding	Specifies the amount of padding to apply to the inside edge of each side of a control.
PressedBrush	Expand this field to select the color for the button when it is pressed.
RenderTransformOrigin	<p>Specifies the center point of any possible render transform, relative to the bounds of the element.</p> <ul style="list-style-type: none"> • X is the horizontal origin factor, given as a value between 0 and 1. • Y is the vertical origin factor, given as a value between 0 and 1.
Row	RowBackground
RowSpan	Specifies the total number of rows that child content spans within a grid.
SnapsToDevicePixels	Select this check box to indicate that rendering for this control uses device-specific pixel settings.
Style	Specifies the style used by this control when it is rendered.
TabIndex	Specifies the order in which controls receive focus when the Tab key is pressed.
Tag	Specifies an arbitrary object value that can be used to store custom information about this control.
Template	Click the ellipsis button to open the Open dialog box to select a template to use.
ToolTip	Specifies the tool-tip that displays for this control in the user interface (UI).

Property	Description
Uid	Specifies the unique identifier (for localization) for this control.
UseLayoutRounding	Specifies a value that determines whether rendering for the object and its visual subtree should use rounding behavior that aligns rendering to whole pixels.
VerticalAlignment	<p>Specifies the vertical alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
VerticalContentAlignment	<p>Specifies the vertical alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
Visibility	<p>Specifies the visibility of this control. Selections are:</p> <ul style="list-style-type: none"> • Collapsed • Hidden • Visible
Width	Specifies the width of the control, in pixels.

Button Controls

A button is a control that causes an event to occur each time it is pressed and released.

Button controls include three categories: regular, repeat, and toggle, with multiple styles in each category. A repeat button causes an event to occur repeatedly from the time it is pressed until it is released. A toggle button can switch between two or three states; that is on/off and on/off/null.

Property	Arrow Button	Flat Toggle Switch	Gel Button	Gel Repeat Button	Gel Toggle Button	Generic Button	Generic Repeat Button	Generic Toggle Button	Industrial Push Button	Industrial Repeat Button	Industrial Toggle Button	Navigator Gel Button	Navigator Generic Button	Power Toggle Button	Rotary Switch State Button	Toggle Switch Button
Value	X		X	X		X			X	X	X	X	X			
Vertical Alignment	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Vertical Content Alignment	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X
Visibility	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Width	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Button Controls Property Descriptions

The following table shows the properties descriptions used by all control button:

Property	Description
AllowDrop	Select this check box to indicate that the control can accept data that is dragged onto it.
ArrowBackground	Expand this field to select the background color for an arrow button control.

Property	Description
ArrowBorderBrush	Expand this field to select the border color of the selected arrow control.
ArrowDesign	<p>Specifies the appearance of the arrow. Available selections are:</p> <ul style="list-style-type: none"> • Pointer • Barbed • Triangle
ArrowDirection	<p>Specifies the direction of the arrow. Available selections are:</p> <ul style="list-style-type: none"> • Up • Down • Right • Left
ArrowPadding	Specifies the horizontal and vertical alignment of the arrow within the button.
Background	Expand this field to select the background color of the selected control.
BorderBrush	Expand this field to select the border color of the selected control.
BorderThickness	Specifies the border thickness of a control, in pixels.
ClickMode	<p>Specifies when the Click event occurs. Selections are:</p> <ul style="list-style-type: none"> • Hover • Press • Release
ClipToBounds	Select this check box to clip the content of this control (or content coming from the child elements of this control) to fit into the size of the containing control.

Property	Description
Column	Specifies the column within the grid where the control is located.
ColumnSpan	Specifies the total number of columns that child content spans within a grid.
CommandParameter	Specifies the parameter to pass to the Command property.
CommandTarget	Specifies the element on which to raise the specified command.
Content	Specifies the content displayed on the control.
ContentStringFormat	Specifies a composite string that determines how to format the Content property if it is displayed as a string.
CornerRadius	Specifies the degree to which the corners of a control are rounded.
Delay	Specifies the amount of time, in milliseconds, that the control waits while being pressed before a command executes.
Effect	<p>Click the arrow to expand the property, select the effect you want to apply to the control, and then configure the specific values for the effect. Valid values to select are:</p> <ul style="list-style-type: none"> • Blur <ul style="list-style-type: none"> ◦ Kernel Type ◦ Radius • Drop Shadow <ul style="list-style-type: none"> ◦ Color ◦ Direction ◦ Opacity ◦ Shadow Depth

Property	Description
FlowDirection	<p>Specifies the direction that text and other user interface (UI) elements flow within any parent control. Selections are:</p> <ul style="list-style-type: none"> • Left to Right • Right to Left
Focusable	<p>Select this check box to indicate that the control can receive focus.</p>
FocusBorderThickness	<p>Specifies the thickness of the focus ring on the outside of the button.</p>
FocusBrush	<p>Expand this field to select the color that displays around the control's border when the control has focus.</p>
FocusVisualStyle	<p>Specifies a property that enables customization of appearance, effects, or other style characteristics that apply to this control when it has focus.</p>
FontFamily	<p>Specifies the font family for the control.</p>
FontSize	<p>Specifies the font size for the control.</p>
FontStretch	<p>Specifies the degree to which a font is condensed or expanded on the screen.</p>
FontStyle	<p>Specifies the font style for the control.</p>
FontWeight	<p>Specifies the weight or thickness of the specified font.</p>
ForceCursor	<p>Select this check box to indicate that the user interface (UI) renders the cursor as declared by the Cursor property.</p>
Foreground	<p>Expand this field to select the foreground color.</p>
GlareBrush	<p>Expand this field to specify the glare effect for the control.</p>
Height	<p>Specifies the height of the control, in pixels.</p>

Property	Description
HorizontalAlignment	<p>Specifies the horizontal alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
HorizontalContentAlignment	<p>Specifies the horizontal alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
Increment	<p>Specifies the amount by which the value is increased each time the button is clicked.</p>
InnerBorderBrush	<p>Expand this field to select the color for the inner border.</p>
InnerBorderThickness	<p>Expand this field to select the color for the inner border.</p>
InnerConrnerRadius	<p>Specifies the degree to which the corners of the inner border are rounded.</p>
Interval	<p>Specifies the amount of time, in milliseconds, between increase or decrease commands when you click a Repeat Button or a Slider.</p>
IsCancel	<p>Select this check box to indicate that a Button or Repeat Button acts as a Cancel button. Press the Esc key to select the actual Cancel button.</p>
IsChecked	<p>Select this check box to indicate that the control is checked.</p>

Property	Description
IsDefault	Select this check box to indicate that a Button or Repeat Button is the default Submit button. Press the Enter key to select the actual Submit button.
IsEnabled	Select this check box to indicate that this control is enabled in the user interface (UI).
IsHitTestVisible	Select this check box to indicate that the control can possibly be returned as a hit test result from some portion of its rendered content.
IsManipulationEnabled	Select this check box to indicate that the manipulation events for the control are enabled.
IsSharedSizeScope	Select this check box to indicate that controls are sharing size information.
IsTabStop	Select this check box to indicate that a control is included in tab navigation.
IsThreeState	Select this check box to indicate that the control supports three states; that is, selected, cleared, indeterminate (the control is selected but dimmed).
Localization	Click the ellipsis button to open the Localization Configuration dialog box where you can configure the property to localize, the language to use for localization, and the text to localize.
Margin	Specifies the outer margin of a control.
MaxHeight	Specifies the maximum height of a control.
MaxWidth	Specifies the maximum width of a control.
MinHeight	Specifies the minimum height of a control.
MinWidth	Specifies the minimum width of a control.
Name	Specifies the identifying name of the control. The name provides a reference so that code behind, such as event handler code, can refer to a markup element during processing by a XAML processor.

Property	Description
OffCaption	Specifies the label to use for the Off position on the control.
OffValue	Select this check box to indicate that the switch on the control starts in the Off position.
OnCaption	Specifies the label to use for the On position on the control.
OnValue	Select this check box to indicate that the switch on the control starts in the On position.
Opacity	Specifies the opacity factor applied to the entire control. Valid values are 0 through 1, with up to 15 decimal places.
OpacityMask	Expand this field to select a color for the opacity mask, which is applied to any alpha-channel masking for the rendered content of this control.
OverridesDefaultStyle	Select this check box to indicate that this control incorporates style properties from theme styles.
Padding	Specifies the amount of padding to apply to the inside edge of each side of a control.
PowerNullBrush	Expand this field to select the color of the button when it is in the third state.
PowerOffBrush	Expand this field to select the color of the button when the power is off.
PowerOnBrush	Expand this field to select the color of the button when the power is on.
PressedBrush	Expand this field to select the color for the button when it is pressed.
RenderTransformOrigin	Specifies the center point of any possible render transform, relative to the bounds of the element.

Property	Description
	<ul style="list-style-type: none"> • X is the horizontal origin factor, given as a value between 0 and 1. • Y is the vertical origin factor, given as a value between 0 and 1.
Row	RowBackground
RowSpan	Specifies the total number of rows that child content spans within a grid.
SnapsToDevicePixels	Select this check box to indicate that rendering for this control uses device-specific pixel settings.
StrokeThickness	Specifies the width of the control's outline.
Style	Specifies the style used by this control when it is rendered.
TabIndex	Specifies the order in which controls receive focus when the Tab key is pressed.
Tag	Specifies an arbitrary object value that can be used to store custom information about this control.
Template	Click the ellipsis button to open the Open dialog box to select a template to use.
ToolTip	Specifies the tool-tip that displays for this control in the user interface (UI).
Uid	Specifies the unique identifier (for localization) for this control.
UseLayoutRounding	Specifies a value that determines whether rendering for the object and its visual subtree should use rounding behavior that aligns rendering to whole pixels.
Value	Specifies the current magnitude of the range control.
VerticalAlignment	Specifies the vertical alignment characteristics applied to this control. Selections are:

Property	Description
	<ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
VerticalContentAlignment	<p>Specifies the vertical alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
Visibility	<p>Specifies the visibility of this control. Selections are:</p> <ul style="list-style-type: none"> • Collapsed • Hidden • Visible
Width	Specifies the width of the control, in pixels.

Arrow Button



An **Arrow Button** control is designed with an arrow in the center. The arrow button causes an event to occur when the button is pressed, and stays pressed until the button is released.

The **Value** property is the default property, and the **Increment Value** property is applied to the **Value** property when the button is clicked

The arrow button has all the properties of a regular button while allowing you to choose an arrow design and direction.

See **Button Controls** to view the usage and description of the properties for this button.

Related reference

[Button Controls \(on page 734\)](#)

Flat Toggle Switch



The **Flat Toggle Switch** is a toggle button with a lever that appears flat. The flat toggle switch control causes an event to occur when it is toggled.

There are three graphical states for the lever: up, down, and neutral. The neutral position appears only when the `IsThreeState` property check box is selected.

See **Button Controls** to view the usage and description of the properties for this button.

Related reference

[Button Controls \(on page 734\)](#)

Gel Button



The **Gel** button control is a generic button with a gel-type look. The gel button causes a single event to occur each time it is pressed and stays active until the button is released.

The **Value** property is the default property, and the **Increment Value** property is applied to the **Value** property when the button is clicked.

See **Button Controls** to view the usage and description of the properties for this button.

Related reference

[Button Controls \(on page 734\)](#)

Gel Repeat Button



The **Gel Repeat** button control is a button with a gel-type look. The gel repeat button causes an event to occur when it is pressed, and repeats the event until the button is released.

The **Value** property is the default property, and the **Increment Value** property is applied to the **Value** property on each event.

See **Button Controls** to view the usage and description of the properties for this button.

Related reference

[Button Controls \(on page 734\)](#)

Gel Toggle Button



The **Gel Toggle** button control is a toggle button with a gel-type appearance. The gel toggle button causes an event to occur when it is toggled.

This control has three possible states: on, off, or null. This button toggles between states each time it is pressed. The button color does not change when the button is toggled; however, the shading changes to indicate the current state. The Null appearance occurs only when the IsThree State property check box is selected.

See **Button Controls** to view the usage and description of the properties for this button.

Related reference

[Button Controls \(on page 734\)](#)

Generic Button



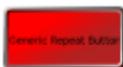
The **Generic** button control causes a single event to occur each time the button is pressed.

See **Button Controls** to view the usage and description of the properties for this control.

Related reference

[Button Controls \(on page 734\)](#)

Generic Repeat Button



The **Generic Repeat** button control causes an event to occur when the button is pressed, and repeats the event until the button is released.

The Value property is the default property, and then Increment Value property is applied to the Value property on each event.

See **Button Controls** to view the usage and description of the properties for this button.

Related reference

[Button Controls \(on page 734\)](#)

Generic Toggle Button

 The **Generic Toggle** button is a toggle button that appears as a flat button. The generic toggle button control causes an event to occur when it is toggled.

This control has three possible states: on, off, or null. This button toggles between states each time it is pressed. The button color changes depending on which state it is in. The Null appearance occurs only when the `IsThreeState` property check box is selected.

See **Button Controls** to view the usage and description of the properties for this button.

Related reference

[Button Controls \(on page 734\)](#)

Industrial Push Button

 The **Industrial Push Button** control is a button with a hexagonal shape. The industrial push button causes a single event to occur each time it is pressed and stays active until the button is released. It has two possible states: on or off.

The `PowerOffBrush` and `PowerOnBrush` properties allow you to set the colors of the button to indicate the state of the button.

See **Button Controls** to view the usage and description of the properties for this button.

Related reference

[Button Controls \(on page 734\)](#)

Industrial Repeat Button

 The **Industrial Repeat Button** control is a button with a hexagonal shape. The industrial repeat button causes an event to occur when the button is pressed, and repeats the event until the button is released.

The `Value` property is the default property, and the `Increment Value` property is applied to the `Value` property on each event.

See **Button Controls** to view the usage and description of the properties for this button.

Related reference

[Button Controls \(on page 734\)](#)

Industrial Toggle Button



The **Industrial Toggle Button control** is a toggle button with a hexagonal shape. The industrial toggle button causes an event to occur when it is toggled. This control has three possible states: on, off, or null. This button toggles between states each time it is pressed.

The Null appearance occurs only when the **IsThreeState** property check box is selected.

See **Button Controls** to view the usage and description of the properties for this button.

Related reference

[Button Controls \(on page 734\)](#)

Navigate Gel Button



The **Navigate Gel Button** is a Silverlight control with a gel-type look. This control is used for navigation to other pages or windows in the application.

The significant properties are `NavigateType` and `NavigateUri`.

See **Button Controls** to view the usage and description of the properties for this button.

Related reference

[Button Controls \(on page 734\)](#)

Navigate Generic Button



The **Navigate Generic Button** is a Silverlight control used for navigation to other pages or windows in the application.

The significant properties are `NavigateType` and `NavigateUri`.

See **Button Controls** to view the usage and description of the properties for this button.

Related reference

[Button Controls \(on page 734\)](#)

Power Toggle Button



The Power Toggle Button control displays the standard power symbol on the button face. This control toggles between On and Off, and the power symbol color changes when the button is pressed.

See **Button Controls** to view the usage and description of the properties for this button.

Related reference

[Button Controls \(on page 734\)](#)

Rotary Switch Three State Button



The Rotary Switch Three State button control is a switch control with a knob. The Rotary Switch Three State button causes an event to occur when the knob is turned. There are three graphical states for the switch: Off, On, and Manual.

See **Button Controls** to view the usage and description of the properties for this button.

Related reference

[Button Controls \(on page 734\)](#)

Toggle Switch Button



The Toggle Switch control is a toggle button with a lever that appears rounded. The toggle switch control causes an event to occur when it is toggled. There are three graphical states for the lever: up, down, and neutral.

The neutral position appears only when the **IsThreeState** property check box is selected.

See **Button Controls** to view the usage and description of the properties for this button.

Related reference

[Button Controls \(on page 734\)](#)

Common Controls

Workflow provides a series of common controls to use in forms and user display authoring.

Common Controls Property Usage

The following table shows the properties used by each control:

Property	Button	Calendar	CheckBox	Expander	Hyperlink	Image	Label	MaskedTextBox	MaskedTextBox	ProgressBar	RadioButton	RepeatButton	RichTextBox	ScrollBar	Slider	TextBlock	TextBox	ToggleButton	ValidationTextBox
Accepts-Return													X			X			X
Accepts-Tab													X			X			X
Allow-Drop	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Auto-ToolTip-Placement															X				
Auto-ToolTip-Precision															X				

Property	Button	Calendar	Check Box	Expander	Hyperlink	Image	Label	Masked Text Box	Class word Box	Progress Bar	Radio Button	Repeat Button	Rich Text Box	Scrollbar	Slider	Text Block	Text Box	Toggle Button	Validating Text Box
Is- Hyphenation- Enabled																X	X	X	
Is- Indeterminate									X										
Is- Manipulation- Enabled	X		X	X		X	X	X	X	X	X	X	X	X	X	X	X		X
Is- Move-To- Point- Enabled															X				
Is- Read-													X				X		X

Property	Button	Calendar	Checkboxpander	Hyper-Image link	Label	Masked Text Box	Class-Progress Bar	Radio Button [add]	Repeat Button	Rich Text Box	Scrollbar	Slider	Text Block	Text Box	Toggle Button	Validating Text Box
Only																
Is-ReadOnly										X				X		X
Only-Caret-Visible																
IsSelection-Range-Enabled												X				
Is-Shared-Size-Scope	X		X		X	X	X	X	X	X	X	X	X	X	X	X
IsSnap-To-Tick-Enabled												X				
Is-Tab	X	X	X	X		X	X	X	X	X	X	X		X	X	X

Property	Button	Calendar	Checkboxpander	Hyper-Image link	Label	Masked Text Box	Class-Progress Bar	Radio Button Item [add]	Repeat Button	Rich Text Box	Scrollbar	Slider	Text Block	Text Box	Toggle Button	Validation Text Box
Stoptop																
Is-Three-Statate								X							X	
Is-Today-Highlighted	X															
Is-Undo-Enabled														X		X
Language	X			X	X											
LargeChange							X				X	X				
Line-Height													X			
Line-Stacking-Style													X			

Property	Button	Calendar	CheckBox Boxpander	Expander	Hyper-Image link	Image	Label	Masked Text Box	Class- word Box	Progress Bar	Radio Button Item [add]	Repeat Button	Rich Text Box	Scroll Bar	Slider	Text Block Box	Text Box	Toggle Button	Validating Text Box
Row-X	X	X			X		X	X	X	X	X	X	X	X	X	X	X	X	X
S-pan																			
Selected- Date	X																		
Selection- Brush								X					X				X		X
Selection- n- End															X				
Selection- Mode	X																		
Selection- Opac- ity								X					X				X		X
Selection- S- tart															X				
Show- Grid- Lines			X																
SmallChange										X				X	X				

Property	Button	Calendar	Checkbox	Expander	Hyperlink	Image	Label	Masked Text Box	Class word Box	Progress Bar	Radio Button	Repeat Button	Rich Text Box	Scrollbar	Slider	Text Block	Text Box	Toggle Button	Invalidating Text Box
Template	X	X		X			X	X	X	X	X	X	X	X	X		X	X	X
Text																X	X		X
TextAlignment																X	X		X
TextTrimming																X			
TextWrapping																X	X		X
TickFrequency															X				
TickPlacement															X				
ToolTip	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Uid	X		X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X
UndoLimit													X				X		X

Property	Button	Calendar	Checkboxpander	Hyper-Image link	Image Label	Masked Text Box	Class-Word Box	Progress Bar	Radio Button Item [add]	Repeat Button	Rich Text Box	Scrollbar	Slider	Text Block	Text Box	Toggle Button	Validating Text Box
Use-Lay-out-Round-ing	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X
Value								X					X				
Ver-ti-cal-Align-ment	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Ver-ti-cal-Con-tent-Al-ign-ment	X	X	X	X		X	X	X	X	X	X	X	X		X	X	X
Ver-ti-cal-Scroll-Bar-Vis-ibil-ity											X				X		X

Property	Button	Calendar	CheckBox	Expander	Hyperlink	Image	Label	MaskedTextBox	MaskedTextBox	ProgressBar	RadioButton	RepeatButton	RichTextBox	ScrollBar	Slider	TextBlock	TextBox	ToggleButton	ValidationTextBox
View-port-Size														X					
Visibility	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Width	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Common Controls Property Descriptions

The following table shows the properties descriptions used by all control:

Property	Description
AcceptsReturn	Select this check box to indicate that pressing Enter in a multiline Text Box control creates a new line of text in the control or activates the default button.
AcceptsTab	Select this check box to indicate that pressing the Tab key in a multiline text box control types a tab character in the control instead of moving the focus to the next control in the tab order.
AllowDrop	Select this check box to indicate that the control can accept data that is dragged onto it.
AutoToolTipPlacement	Specifies whether a tooltip that contains the current value of the slider displays when the thumb is pressed and also specifies the placement of the tooltip. Selections are:

Property	Description
	<ul style="list-style-type: none"> • None • Top Left • Bottom Right
AutoToolTipPrecision	Specifies the number of digits to the right side of the decimal point to display in a tooltip for the value of the slider.
AutoWordSelection	Select this check box to indicate that when you select part of a word by dragging across it with the mouse, the rest of the word is selected.
ArrowBackground	Expand this field to select the background color for an arrow button control.
ArrowBorderBrush	Expand this field to select the border color of the selected arrow control.
ArrowDesign	<p>Specifies the appearance of the arrow. Available selections are:</p> <ul style="list-style-type: none"> • Pointer • Barbed • Triangle
ArrowDirection	<p>Specifies the direction of the arrow. Available selections are:</p> <ul style="list-style-type: none"> • Up • Down • Right • Left
ArrowPadding	Specifies the horizontal and vertical alignment of the arrow within the button.
Background	Expand this field to select the background color of the selected control.
BaselineOffset	Specifies the amount of space, by which each line of text is offset from the baseline.

Property	Description
BorderBrush	Expand this field to select the border color of the selected control.
BorderThickness	Specifies the border thickness of a control, in pixels.
CacheMode	Specifies a value that indicates that rendered content should be cached when possible.
CalendarButtonStyle	Specifies the style of the calendar's buttons.
CalendarDayButtonStyle	Specifies the style of the calendar's day buttons.
CalendarItemStyle	Specifies the style of the calendar's items.
CaretBrush	Expand this field to select the brush to use to render the vertical bar that indicates the insertion point.
CharacterCasing	<p>Specifies how characters are cased when they are manually entered into the text box. Selections are:</p> <ul style="list-style-type: none"> • Normal • Lower • Upper
ClickMode	<p>Specifies when the Click event occurs. Selections are:</p> <ul style="list-style-type: none"> • Hover • Press • Release
Clip	<p>Specifies the geometric shape used to define the outline of the image. Selections are:</p> <ul style="list-style-type: none"> • Hover • Press • Release
ClipToBounds	Select this check box to clip the content of this control (or content coming from the child elements

Property	Description
	of this control) to fit into the size of the containing control.
Column	Specifies the column within the grid where the control is located.
ColumnSpan	Specifies the total number of columns that child content spans within a grid.
Command	Specifies the command to invoke when this control is pressed or clicked.
CommandParameter	Specifies the parameter to pass to the Command property.
CommandTarget	Specifies the element on which to raise the specified command.
Content	Specifies the content displayed on the control.
ContentStringFormat	Specifies a composite string that determines how to format the Content property if it is displayed as a string.
ContentTemplate	Specifies the data template used to display the content of the Content property.
Cursor	Specifies the cursor image to display when the mouse pointer hovers over an object.
DataContext	Specifies the data context for a control when it participates in data binding.
Delay	Specifies the amount of time, in milliseconds, that the control waits while being pressed before a command executes.
DisplayDate	Specifies the date to display.
DisplayDateEnd	Specifies the last date in the date range that is available in the calendar.
DisplayDateStart	Specifies the first date that is available in the calendar.

Property	Description
DisplayMode	Specifies whether the calendar displays a month, year, or decade.
Effect	<p>Click the arrow to expand the property, select the effect you want to apply to the control, and then configure the specific values for the effect. Valid values to select are:</p> <ul style="list-style-type: none"> • Blur <ul style="list-style-type: none"> ◦ Kernel Type ◦ Radius • Drop Shadow <ul style="list-style-type: none"> ◦ Color ◦ Direction ◦ Opactiy ◦ Shadow Depth
FirstDayOfWeek	<p>Specifies the day that is considered the beginning of the week. Selections are:</p> <ul style="list-style-type: none"> • Sunday • Modnay • Tuesday • Wednesday • Thursday • Friday • Saturday
FlowDirection	<p>Specifies the direction that text and other user interface (UI) elements flow within any parent control. Selections are:</p> <ul style="list-style-type: none"> • Left to Right • Right to Left
Focusable	Select this check box to indicate that the control can receive focus.

Property	Description
FocusVisualStyle	Specifies a property that enables customization of appearance, effects, or other style characteristics that apply to this control when it has focus.
FontFamily	Specifies the font family for the control.
FontSize	Specifies the font size for the control.
FontStretch	Specifies the degree to which a font is condensed or expanded on the screen.
FontStyle	Specifies the font style for the control.
FontWeight	Specifies the weight or thickness of the specified font.
ForceCursor	Select this check box to indicate that the user interface (UI) renders the cursor as declared by the Cursor property.
Foreground	Expand this field to select the foreground color.
Height	Specifies the height of the control, in pixels.
HorizontalAlignment	<p>Specifies the horizontal alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
HorizontalContentAlignment	<p>Specifies the horizontal alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
HorizontalScrollBarVisibility	<p>Specifies whether a horizontal scroll bar is shown. Selections are:</p>

Property	Description
	<ul style="list-style-type: none"> • Auto - the scroll bar appears only if the content exceeds the width of the control • Disabled - the scroll bar never appears • Hidden - the scroll bar is hidden from view • Visible - the scroll bar is visible at all times
ImageStretch	<p>Specifies how the image should be stretched to fill the destination rectangle. Selections are:</p> <ul style="list-style-type: none"> • Fill • None • Uniform • UniformToFill
Interval	<p>Specifies the amount of time, in milliseconds, between increase or decrease commands when you click a Repeat Button or a Slider.</p>
IsCancel	<p>Select this check box to indicate that a Button or Repeat Button acts as a Cancel button. Press the Esc key to select the actual Cancel button.</p>
IsChecked	<p>Select this check box to indicate that the control is checked.</p>
IsDefault	<p>Select this check box to indicate that a Button or Repeat Button is the default Submit button. Press the Enter key to select the actual Submit button.</p>
IsDirectionReversed	<p>Select this check box to indicate that the direction of increasing value is reversed; that is, from right to left.</p>
IsDocumentEnabled	<p>Select this check box to indicate that you can interact with UIElement and ContentElement objects within the Rich Text Box.</p>
IsEnabled	<p>Select this check box to indicate that this control is enabled in the user interface (UI).</p>

Property	Description
IsHitTestVisible	Select this check box to indicate that the control can possibly be returned as a hit test result from some portion of its rendered content.
IsHyphenationEnabled	Select this check box to indicate that automatic hyphenation of words is enabled.
IsIndeterminate	Select this check box to indicate that no actual values display on the progress bar. Clear this check box to display actual values.
IsManipulationEnabled	Select this check box to indicate that the manipulation events for the control are enabled.
IsMoveToPointEnabled	Select this check box to indicate that the thumb of a slider moves immediately to the location of the mouse click that occurs while the mouse pointer pauses on the Slider track.
IsReadOnly	Select this check box to indicate that the text editing control is read-only to a user interacting with the control.
IsReadOnlyCaretVisible	Specifies whether a read-only text box displays a caret. If the IsReadOnly check box is selected, you can still select and copy text. If the IsReadOnlyCaretVisible check box is also selected, a caret appears in the text box when the text box has keyboard focus. When the IsReadOnly check box is cleared, selecting the IsReadOnlyCaretVisible check box has no effect.
IsSelectionRangeEnabled	Select this check box to indicate that the Slider displays a selection range.
IsSharedSizeScope	Select this check box to indicate that controls are sharing size information.
IsSnapToTickEnabled	Select this check box to indicate that the Slider automatically moves the thumb to the closest tick mark.

Property	Description
IsTabStop	Select this check box to indicate that a control is included in tab navigation.
IsThreeState	Select this check box to indicate that the control supports three states; that is, selected, cleared, indeterminate (the control is selected but dimmed).
IsTodayHighlighted	Select this check box to indicate that the current date is highlighted.
IsUndoEnabled	Select this check box to indicate that undo support is enabled for the text-editing control. Clearing this check box clears the undo stack. Therefore, if you disable undo and then re-enable it, undo commands still do not work because the undo stack was emptied when you cleared the check box.
Language	Specifies the localization language that applies to an object.
LargeChange	Specifies a value to be added to or subtracted from the value of a range base control.
LineHeight	Specifies the height of each line of content.
LineStackingStrategy	<p>Specifies the mechanism by which a line box is determined for each line of text within the Text Block. Selections are:</p> <ul style="list-style-type: none"> • BlockLineHeight • MaxHeight
Localization	Click the ellipsis button to open the Localization Configuration dialog box where you can configure the property to localize, the language to use for localization, and the text to localize.
Margin	Specifies the outer margin of a control.
MaxHeight	Specifies the maximum height of a control.
MaxLength	Specifies the maximum length of text that can be entered in the control. When this property is set

Property	Description
	to 0, the maximum length of the text that can be entered in the control is limited only by available memory.
MaxLines	Specifies the maximum number of visible lines.
MaxWidth	Specifies the maximum width of a control.
MinHeight	Specifies the minimum height of a control.
MinLines	Specifies the minimum number of visible lines.
MinWidth	Specifies the minimum width of a control.
Name	Specifies the identifying name of the control. The name provides a reference so that code behind, such as event handler code, can refer to a markup element during processing by a XAML processor.
NavigateUri	Specifies the URI to navigate to when the control is clicked.
Opacity	Specifies the opacity factor applied to the entire control. Valid values are 0 through 1, with up to 15 decimal places.
OpacityMask	Expand this field to select a color for the opacity mask, which is applied to any alpha-channel masking for the rendered content of this control.
Orientation	<p>Specifies the orientation of a control. Selections are:</p> <ul style="list-style-type: none"> • Portrait • Landscape
OverridesDefaultStyle	Select this check box to indicate that this control incorporates style properties from theme styles.
Padding	Specifies the amount of padding to apply to the inside edge of each side of a control.
PasswordChar	Specifies the masking character for the Password Box.

Property	Description
Projection	Specifies the 3-D effect to apply to the current object.
RenderTransformOrigin	<p>Specifies the center point of any possible render transform, relative to the bounds of the element.</p> <ul style="list-style-type: none"> • X is the horizontal origin factor, given as a value between 0 and 1. • Y is the vertical origin factor, given as a value between 0 and 1.
Row	RowBackground
RowSpan	Specifies the total number of rows that child content spans within a grid.
SelectedDate	Specifies the currently selected date.
SelectionBrush	Expand this field to select the brush to use to highlight the selected text.
SelectionEnd	Specifies the largest value of a specified selection for a slider control. The value of this property cannot be greater than the value of the Maximum property and cannot be less than the value of the Minimum property. The value of this property must also be greater than or equal to the value of the SelectionStart property. If the value of this property is greater than the value of the Maximum property or less than the value of the Minimum property, the value is set to the value of the Maximum or Minimum property respectively.
SelectionMode	<p>Specifies the selection behavior for a List Box or List View control. Selections are:</p> <ul style="list-style-type: none"> • Single Date • Single Range • Multiple Range • None

Property	Description
SelectionStart	Specifies the smallest value of a specified selection for a slider control. The value of this property cannot be less than the value of the Minimum property and cannot be greater than the value of the Maximum property. The value of this property must also be greater than or equal to the value of the SelectionEnd property. If the value of this property is less than the value of the Minimum property or greater than the value of the Maximum property, the value is set to the value of the Minimum or Maximum property respectively.
SelectionOpacity	Specifies the opacity factor applied to the selected text. Valid values are 0 through 1, with up to 15 decimal places. If the value is set to 0, the SelectionBrush is transparent and is not visible. If the value is set to 1.0 or greater, the SelectionBrush is opaque and the selected text is not visible.
ShowGridLines	Select this check box to indicate that grid lines are visible within this grid.
SmallChange	Specifies a value to be added to or subtracted from the value of a range base control.
SnapsToDevicePixels	Select this check box to indicate that rendering for this control uses device-specific pixel settings.
Source	Click the ellipsis button to open the Select an Image dialog box, where you can select the image to use for the control.
Stretch	<p>Specifies how an Image stretches to fill the destination rectangle. Selections are:</p> <ul style="list-style-type: none"> • Fill • None (does not apply to Scale Box) • Uniform (does not apply to Scale Box) • UniformFill (does not apply to Scale Box)

Property	Description
StretchDirection	<p>Specifies how the image is scaled. Selections are:</p> <ul style="list-style-type: none"> • Vertical • Horizontal
Style	<p>Specifies the style used by this control when it is rendered.</p>
TabIndex	<p>Specifies the order in which controls receive focus when the Tab key is pressed.</p>
TabNavigation	<p>Specifies a value that modifies how tabbing and the TabIndex property work for this control.</p>
Tag	<p>Specifies an arbitrary object value that can be used to store custom information about this control.</p>
TargetName	<p>Specifies the name of the target window or frame that the web page should open in, or the name of the object within the Silverlight application to navigate to.</p>
Template	<p>Click the ellipsis button to open the Open dialog box to select a template to use.</p>
Text	<p>Specifies the text contents of a control.</p>
TextAlignment	<p>Specifies the horizontal alignment of text content. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Justify
TextTrimming	<p>Specifies the text trimming behavior to employ when content overflows the content area. Selections are:</p> <ul style="list-style-type: none"> • None • CharacterEllipsis • WordEllipsis

Property	Description
TextWrapping	<p>Specifies how the control wraps text. Selections are:</p> <ul style="list-style-type: none"> • NoWrap • Wrap • WrapWithOverflow
TickFrequency	Specifies the interval between tick marks.
TickPlacement	<p>Specifies the position of tick marks with respect to the track of the Slider. Selections are:</p> <ul style="list-style-type: none"> • None • Both • TopLeft • BottomRight
ToolTip	Specifies the tool-tip that displays for this control in the user interface (UI).
Uid	Specifies the unique identifier (for localization) for this control.
UndoLimit	Specifies the number of actions stored in the undo queue.
UseLayoutRounding	Specifies a value that determines whether rendering for the object and its visual subtree should use rounding behavior that aligns rendering to whole pixels.
Value	Specifies the current magnitude of the range control.
VerticalAlignment	<p>Specifies the vertical alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch

Property	Description
VerticalContentAlignment	<p>Specifies the vertical alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
VerticalScrollBarVisibility	<p>Specifies whether a vertical scroll bar is shown. Scroll bar selections are:</p> <ul style="list-style-type: none"> • Auto - appears only if the content exceeds the width of the control. • Disabled - never appears. • Hidden - hidden from view. • Visible - visible at all times.
ViewportSize	<p>Specifies the amount of the scrollable content that is currently visible. The value of this property is used to calculate the size of the thumb control that is displayed as the sliding value indicator in a scrollbar control. The thumb control's size represents the amount of a scrollviewer control's content that is visible. For example, if 25 percent of a scrollviewer control's content is visible, the thumb control occupies 25 percent of the track in the scrollbar.</p>
Visibility	<p>Specifies the visibility of this control. Selections are:</p> <ul style="list-style-type: none"> • Collapsed • Hidden • Visible
Width	<p>Specifies the width of the control, in pixels.</p>

Add content to an Expander control

The Expander control is used with a Panel or Grid control.

Before you begin



Note:

Ensure that Expander control **Height** property is set to Auto.

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the Toolbox tab.
5. Expand the control group you want to work with.
6. Click and drag the control to the desired position in the **Designer**.



Note:

To apply the properties of a control to another control, click and drag the control to another control in the document tree. The Applying <control (control name)> to Target dialog box appears where you can select the option you want to use to apply to the control.

7. Add a grid or panel.
8. Right-click the control in either the Designer or the document tree, and then select **Edit Layer**.

- Or -

Double-click the control in the document tree.

The control displays in the designer with an orange border, indicating that it is the active layer.

9. Add the appropriate controls to the grid or panel.

When you make another control the active layer, you cannot select the controls that were added in other active layers because they are now part of an inactive layer. In order to change these controls, you must make that layer active again.

10. Reset the top level control to be the active layer.
11. Right-click the grid/panel control, and then select **Cut**.
12. In the **Document Tree**, right-click the **Expander** control, and then select **Paste Special**.

The Applying <control> to Target Expander dialog box appears.

13. Select the **Child of Expander** option, and then click **OK**.

When the project is built, you can click the drop-down arrow on the expander to display the grid/panel.

**Important:**

You can add only one grid or panel control to an expander. If a grid or panel control already exists as content in your expander control, adding a new grid or panel overwrites the existing one.

14. Configure the properties, as required.

**Note:**

To view the full list of properties, select the **Advanced** check box.

15. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Add an image

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Document Tree**.
4. Select the image control and, in the **Properties** panel, select the ellipsis button for the **Source** property.
The Navigate Uri dialog box appears.
5. Select a local or remote image source.
6. Configure the properties, as required.

**Note:**

To view the full list of properties, select the **Advanced** check box.

7. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Related information

[Add a local source file \(on page 1102\)](#)

[Add a remote source file \(on page 1102\)](#)

Button

 A Button is a control you can click to provide input to an application. A button's appearance can change by adjusting its background, foreground, or size, but its behavior can also change if adjustments are made to the actions connected to the click event.

A click event is what happens to the button when it is clicked. For example, by changing the button's ClickMode property from *release* to *hover*, hovering the mouse cursor over the button (as opposed to clicking and releasing the button) produces the intended action.

The Button is a content control; to modify its text or content, use the Content property.

See **Common Controls** to view the usage and description of the properties for this control.

Related reference

[Common Controls \(on page 758\)](#)

Calendar



The Calendar control is a Silverlight control that allows you to select a date by using a visual calendar display.

See **Common Controls** to view the usage and description of the properties for this control.

Related reference

[Common Controls \(on page 758\)](#)

Check Box



Check Box controls are used for groups of non-exclusive items, where more than one item in the group can be selected at the same time.

A check box's appearance and behavioral properties can be changed. The Check Box is a content control; to modify its text or content, use the Content property.

See **Common Controls** to view the usage and description of the properties for this control.

Related reference

[Common Controls \(on page 758\)](#)

Expander



An Expander is a control that organizes objects by providing a heading and the ability to expand or collapse the contents.

SectionTitle

An expander's Header property is used for the title and can be a string or other WPF element. The contents of an expander are configured in the Content property and are usually a Panel or Grid with children.



Important:

The Expander control may not work as desired if its Height property is specified. Ensure the Height property is set to Auto.

See **Common Controls** to view the usage and description of the properties for this control.

Related reference

[Common Controls \(on page 758\)](#)

Hyperlink



The Hyperlink is a Silverlight control that allows you to add a link to navigate to a web page or other Silverlight object.

See **Common Controls** to view the usage and description of the properties for this control.

Related reference

[Common Controls \(on page 758\)](#)

Image



An Image control allows you to add pre-designed graphic images or pictures to your form.

The Source property allows you to access the image you want to use. You can use images that have been added to the Resources folder on the Project panel, or you can use remote images. For more information, see [Navigate URI \(on page 1099\)](#).

The Source property allows you to access the image you want to use. You can only use images that have been added to the document repository and that have a document type of *Image*.

See **Common Controls** to view the usage and description of the properties for this control.

Related reference

[Common Controls \(on page 758\)](#)

Label



A Label control is used for labeling objects in your form.

A Label is a content control. The Content property accepts a string or other WPF element as content. For web applications, a Text Block control can be used in place of a label.

See **Common Controls** to view the usage and description of the properties for this control.

Related reference

[Common Controls \(on page 758\)](#)

Masked Text Box



A Masked Text Box control uses a mask to distinguish between proper and improper user input.

Using a masked text box, you can specify the following input without writing any custom validation:

- Required input characters.
- Optional input characters.

- The type of input expected at a given position in the mask; for example, a digit, or an alphabetic or alphanumeric character.
- Mask literals, or characters that should appear directly in the MaskedTextBox control; for example, the hyphens (-) in a phone number, or the currency symbol in a price.
- Special processing for input characters; for example, to convert alphabetic characters to uppercase.

See **Common Controls** to view the usage and description of the properties for this control.

Related reference

[Common Controls \(on page 758\)](#)

Password Box



A Password Box control is a type of text box used specifically to create a password control.

In addition to changing the appearance and behavior of the password box, you can set the number of characters for and appearance of the password.

See **Common Controls** to view the usage and description of the properties for this control.

Related reference

[Common Controls \(on page 758\)](#)

Progress Bar



The Progress Bar is a control with built-in, looped animation and is used to show that an operation is in progress. An animated progress bar indicates that a user must wait before performing another task.

By default, the `IsIndeterminate` property check box is cleared. Selecting the `IsIndeterminate` property check box allows you to enable animation on the progress bar.

Other commonly used properties for the Progress Bar include `Minimum`, `Maximum`, and `Value`.

See **Common Controls** to view the usage and description of the properties for this control.

Related reference

[Common Controls \(on page 758\)](#)

Radio Button Item [add]



Radio Button controls are used to select a single option from a group of choices when paired with other radio buttons. Unlike check boxes, radio buttons are used for groups where an operator must make an either/or choice from a group of mutually exclusive options. Within the group, no two radio buttons can be selected at one time.

A Radio Button control accepts either text or a child object as content. To add text as content, use the Content property.

In order for a number of radio buttons to function together as a mutually exclusive group, they must be given a group name. Radio buttons with the same GroupName property function together as a mutually exclusive radio button list. Radio buttons possess an IsChecked property that specifies whether the radio button is checked.

See **Common Controls** to view the usage and description of the properties for this control.

Related reference

[Common Controls \(on page 758\)](#)

Repeat Button



The Repeat Button triggers its click event repeatedly while it is pressed.

Commonly used properties on this control are Interval and Delay.

See **Common Controls** to view the usage and description of the properties for this control.

Related reference

[Common Controls \(on page 758\)](#)

Rich Text Box



A Rich Text Box is an enhanced text box control that provides advanced formatting features.

For example, using keyboard shortcuts, you can apply formatting to selected text during run time.

See **Common Controls** to view the usage and description of the properties for this control.

Related reference

[Common Controls \(on page 758\)](#)

Scroll Bar

 The Scroll Bar control is used to facilitate scrolling; for example, through a list of items or a document with substantial content. Scroll bars can have either horizontal or vertical orientation.

Common properties used on the scroll bar are Minimum, Maximum, Value, and Orientation.

See **Common Controls** to view the usage and description of the properties for this control.

Related reference

[Common Controls \(on page 758\)](#)

Slider

 The Slider control enables you to select a value from a range of values, usually from a minimum to a maximum, along a continuum. For example, the volume on a stereo. The left-hand side usually represents minimum value and the right-hand side usually represents high (the maximum) value, with a range of values in between.

The Slider control has a horizontal or vertical orientation, a track, and a movable thumb. A slider's track can be annotated to show starting and ending points and ticks for the different values along the continuum. Unlike a scroll bar, a slider's thumb remains fixed – both in size and in relation to the value it represents.

Common properties used on the slider are Minimum, Maximum, Value, and Orientation.

See **Common Controls** to view the usage and description of the properties for this control.

Related reference

[Common Controls \(on page 758\)](#)

Text Block

 The Text Block control is used for entering and editing blocks of text. This control has properties for multi-line text wrapping and alignment, which makes it effective for the layout of larger sections of text.

Setting text on a text block is done using the Text property. This control is often used in place of a Label control for web and non-web applications.

See **Common Controls** to view the usage and description of the properties for this control.

Related reference

[Common Controls \(on page 758\)](#)

Text Box



The Text Box control is used for entering and editing alphanumeric text. This is a very basic control as it accepts only text as content.

Unlike the single line capability of a Label control, however, a text box has an AcceptsReturn property that allows text to appear on multiple lines. In addition, a text box has a TextWrapping property that provides options for automatically wrapping text within the box's parameters.

Setting text on a text box is done using the Text property.

See **Common Controls** to view the usage and description of the properties for this control.

Related reference

[Common Controls \(on page 758\)](#)

Toggle Button



The Toggle Button control differs from Button or Repeat Button controls in that it toggles on and off.

It has an IsChecked property, which makes the button appear pressed and held down, and an IsThreeState property, which determines whether the control supports two or three states. If a toggle button is configured to have three states, it allows the operator to choose a third, indeterminate state. For example, you could use a three-state toggle button to indicate Yes, No, or Not Applicable.

See **Common Controls** to view the usage and description of the properties for this control.

Related reference

[Common Controls \(on page 758\)](#)

Validating Text Box

 The Validating Text Box control allows you to add a text box to your project that requires a valid entry. The validating text box displays a colored border until a valid value is entered.

See **Common Controls** to view the usage and description of the properties for this control.

Related reference

[Common Controls \(on page 758\)](#)

Container Controls

Container controls can contain a single child control that is manipulated by the parent control.

Container Controls Property Usage

The following table shows the properties used by each control:

Property	Blow Out Border	Scale Box
AllowDrop	X	X
AnimationSpeed	X	
Background	X	X
BorderBrush	X	X
BorderThickness	X	X
ClipToBounds	X	X
Column	X	X
ColumnSpan	X	X
Content	X	X
ContentStringFormat	X	X
Effect	X	X
FlowDirection	X	X
Focusable	X	X
FocusVisualStyle	X	X
FontFamily	X	X

Property	Blow Out Border	Scale Box
FontSize	X	X
FontStretch	X	X
FontStyle	X	X
FontWeight	X	X
ForceCursor	X	X
Foreground	X	X
Height	X	X
HorizontalAlignment	X	X
HorizontalContentAlignment	X	X
IsEnabled	X	X
IsHitTestVisible	X	X
IsManipulationEnabled	X	X
IsSharedSizeScope	X	X
IsTabStop	X	X
Localization	X	X
Margin	X	X
MaxHeight	X	X
MaxWidth	X	X
MinHeight	X	X
MinWidth	X	X
Name	X	X
Opacity	X	X
OpacityMask	X	X
OverridesDefaultStyle	X	X
Padding	X	X
RenderTransformOrigin	X	X

Property	Blow Out Border	Scale Box
Row	X	X
RowSpan	X	X
Scale	X	
SnapsToDevicePixels	X	X
Stretch		X
StretchDirection		X
Style	X	X
TabIndex	X	X
Tag	X	X
Template	X	X
ToolTip	X	X
Uid	X	X
UseLayoutRounding	X	X
VerticalAlignment	X	X
VerticalContentAlignment	X	X
Visibility	X	X
Width	X	X

Container Controls Property Descriptions

The following table shows the properties descriptions used by all control:

Property	Description
AllowDrop	Select this check box to indicate that the control can accept data that is dragged onto it.
AnimationSpeed	Specifies the speed, as a numeric value, of the control's content.
Background	Expand this field to select the background color of the selected control.

Property	Description
BorderBrush	Expand this field to select the border color of the selected control.
BorderThickness	Specifies the border thickness of a control, in pixels.
ClipToBounds	Select this check box to clip the content of this control (or content coming from the child elements of this control) to fit into the size of the containing control.
Column	Specifies the column within the grid where the control is located.
ColumnSpan	Specifies the total number of columns that child content spans within a grid.
Content	Specifies the content displayed on the control.
ContentStringFormat	Specifies a composite string that determines how to format the Content property if it is displayed as a string.
Effect	<p>Click the arrow to expand the property, select the effect you want to apply to the control, and then configure the specific values for the effect. Valid values to select are:</p> <ul style="list-style-type: none"> • Blur <ul style="list-style-type: none"> ◦ Kernel Type ◦ Radius • Drop Shadow <ul style="list-style-type: none"> ◦ Color ◦ Direction ◦ Opacity ◦ Shadow Depth
FlowDirection	Specifies the direction that text and other user interface (UI) elements flow within any parent control. Selections are:

Property	Description
	<ul style="list-style-type: none"> • Left to Right • Right to Left
Focusable	Select this check box to indicate that the control can receive focus.
FocusVisualStyle	Specifies a property that enables customization of appearance, effects, or other style characteristics that apply to this control when it has focus.
FontFamily	Specifies the font family for the control.
FontSize	Specifies the font size for the control.
FontStretch	Specifies the degree to which a font is condensed or expanded on the screen.
FontStyle	Specifies the font style for the control.
FontWeight	Specifies the weight or thickness of the specified font.
ForceCursor	Select this check box to indicate that the user interface (UI) renders the cursor as declared by the Cursor property.
Foreground	Expand this field to select the foreground color.
Height	Specifies the height of the control, in pixels.
HorizontalAlignment	<p>Specifies the horizontal alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
HorizontalContentAlignment	<p>Specifies the horizontal alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Left • Center

Property	Description
	<ul style="list-style-type: none"> • Right • Stretch
IsEnabled	Select this check box to indicate that this control is enabled in the user interface (UI).
IsHitTestVisible	Select this check box to indicate that the control can possibly be returned as a hit test result from some portion of its rendered content.
IsManipulationEnabled	Select this check box to indicate that the manipulation events for the control are enabled.
IsSharedSizeScope	Select this check box to indicate that controls are sharing size information.
IsTabStop	Select this check box to indicate that a control is included in tab navigation.
Localization	Click the ellipsis button to open the Localization Configuration dialog box where you can configure the property to localize, the language to use for localization, and the text to localize.
Margin	Specifies the outer margin of a control.
MaxHeight	Specifies the maximum height of a control.
MaxWidth	Specifies the maximum width of a control.
MinHeight	Specifies the minimum height of a control.
MinWidth	Specifies the minimum width of a control.
Name	Specifies the identifying name of the control. The name provides a reference so that code behind, such as event handler code, can refer to a markup element during processing by a XAML processor.
Opacity	Specifies the opacity factor applied to the entire control. Valid values are 0 through 1, with up to 15 decimal places.

Property	Description
OpacityMask	Expand this field to select a color for the opacity mask, which is applied to any alpha-channel masking for the rendered content of this control.
OverridesDefaultStyle	Select this check box to indicate that this control incorporates style properties from theme styles.
Padding	Specifies the amount of padding to apply to the inside edge of each side of a control.
RenderTransformOrigin	<p>Specifies the center point of any possible render transform, relative to the bounds of the element.</p> <ul style="list-style-type: none"> • X is the horizontal origin factor, given as a value between 0 and 1. • Y is the vertical origin factor, given as a value between 0 and 1.
Row	RowBackground
RowSpan	Specifies the total number of rows that child content spans within a grid.
Scale	Specifies, by a numeric multiple, the size to which the content is enlarged.
SnapsToDevicePixels	Select this check box to indicate that rendering for this control uses device-specific pixel settings.
Stretch	<p>Specifies how an Image stretches to fill the destination rectangle. Selections are:</p> <ul style="list-style-type: none"> • Fill • None (does not apply to Scale Box) • Uniform (does not apply to Scale Box) • UniformFill (does not apply to Scale Box)
StretchDirection	<p>Specifies how the image is scaled. Selections are:</p> <ul style="list-style-type: none"> • Vertical • Horizontal

Property	Description
Style	Specifies the style used by this control when it is rendered.
TabIndex	Specifies the order in which controls receive focus when the Tab key is pressed.
Tag	Specifies an arbitrary object value that can be used to store custom information about this control.
Template	Click the ellipsis button to open the Open dialog box to select a template to use.
ToolTip	Specifies the tool-tip that displays for this control in the user interface (UI).
Uid	Specifies the unique identifier (for localization) for this control.
UseLayoutRounding	Specifies a value that determines whether rendering for the object and its visual subtree should use rounding behavior that aligns rendering to whole pixels.
VerticalAlignment	<p>Specifies the vertical alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
VerticalContentAlignment	<p>Specifies the vertical alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
Visibility	Specifies the visibility of this control. Selections are:

Property	Description
	<ul style="list-style-type: none"> • Collapsed • Hidden • Visible
Width	Specifies the width of the control, in pixels.

Blow Out Border



The Blow Out Border is a WPF control that magnifies its content when the mouse pointer moves over the control.

The Scale property is used to display different percentages of magnification. The blow out border control magnifies the content as it appears inside the border; it does not change the actual size of the content.



Note:

Content is added as a child to the blow out border control.

See **Container Controls** to view the usage and description of the properties for this control.

Related reference

[Container Controls \(on page 800\)](#)

Scale Box



The Scale Box is a WPF control that scales its content relative to the proportion of its bounding box size.

Content that is to be resized is set as a child of the scale box. The scale box and its child content can be resized in the designer only; you cannot resize the child control in run time.

See **Container Controls** to view the usage and description of the properties for this control.

Related reference

[Container Controls \(on page 800\)](#)

Date and Time Controls

Date and time controls allow you to add clocks and interactive calendars to a display.

Common Controls Property Usage

The following table shows the properties used by each control:

Property	Analog Clock	Date Time Display	Date/Time Picker	Duration Picker
AllowDrop	X	X	X	X
Background	X	X	X	X
BorderBrush	X	X	X	X
BorderThickness	X	X	X	X
CenterCapBorder-Brush	X			
CenterCapBrush	X			
ClipToBounds		X	X	X
Column	X	X	X	X
ColumnSpan	X	X	X	X
CornerRadius		X		
DateTimeFormat		X		
DateTimeFormats		X		
DisplayLongFormat			X	
Duration				X
Effect	X	X	X	X
FlowDirection	X	X	X	X
Focusable	X	X	X	X
FocusVisualStyle	X	X	X	X
FontFamily	X	X	X	X
FontSize	X	X	X	X
FontStretch	X	X	X	X
FontStyle	X	X	X	X
FontWeight	X	X	X	X

Property	Analog Clock	Date Time Display	Date/Time Picker	Duration Picker
ForceCursor	X	X	X	X
Foreground	X	X	X	X
GlareBrush	X	X		
HasError			X	X
Height	X	X	X	X
HorizontalAlign- ment	X	X	X	X
HorizontalContent- Alignment	X	X	X	X
InnerBorderBrush		X		
InnerBorderThick- ness		X		
InnerCornerRadius		X		
IsEnabled	X	X	X	X
IsHitTestVisible	X	X	X	X
IsManipulationEn- abled	X	X	X	X
IsSharedSizeScope	X	X	X	X
IsTabStop	X	X	X	X
Localization	X	X	X	X
Margin	X	X	X	X
MaxHeight	X	X	X	X
MaxWidth	X	X	X	X
MinHeight	X	X	X	X
MinWidth	X	X	X	X
Name	X	X	X	X
NeedleBrush	X			

Property	Analog Clock	Date Time Display	Date/Time Picker	Duration Picker
Opacity	X	X	X	X
OpacityMask	X	X	X	X
OverridesDefault- Style	X	X	X	X
Padding	X	X	X	X
RenderTransform- Origin	X	X	X	X
Row	X	X	X	X
RowSpan	X	X	X	X
SecondsNeedle- Brush	X			
SelectedDateTime			X	
ShowMilliseconds			X	X
SnapsToDevicePix- els	X	X	X	X
Style	X	X	X	X
TabIndex	X	X	X	X
Tag	X	X	X	X
Template	X	X	X	X
TickBorderBrush	X			
TickBrush	X			
ToolTip	X	X	X	X
Uid	X	X	X	X
UseLayoutRound- ing	X	X	X	X
VerticalAlignment	X	X	X	X
VerticalContentAl- ignment	X	X	X	X

Property	Analog Clock	Date Time Display	Date/Time Picker	Duration Picker
Visibility	X	X	X	X
Width	X	X	X	X

Common Controls Property Descriptions

The following table shows the properties descriptions used by all control:

Property	Description
AllowDrop	Select this check box to indicate that the control can accept data that is dragged onto it.
Background	Expand this field to select the background color of the selected control.
BorderBrush	Expand this field to select the border color of the selected control.
BorderThickness	Specifies the border thickness of a control, in pixels.
CenterCapBorderBrush	Expand this field to select a color for the outside edge of the center cap.
CenterCapBrush	Expand this field to select a color for the center cap.
ClipToBounds	Select this check box to clip the content of this control (or content coming from the child elements of this control) to fit into the size of the containing control.
Column	Specifies the column within the grid where the control is located.
ColumnSpan	Specifies the total number of columns that child content spans within a grid.
Content	Specifies the content displayed on the control.
CornerRadius	Specifies the degree to which the corners of a control are rounded.

Property	Description
DateTimeFormat	Specifies the custom date/time display format.
DateTimeFormats	<p>Specifies the date/time display format. Available selections are:</p> <ul style="list-style-type: none"> • Custom • Day Month • Default Date Time • Default Date Time Long • Full Date Time • Full Date Time Long • Hour Minute Second • Long Date • Long Time • Month Day • RFC1123Date • Short Date • Short Time • Sortable Date • Universal Local Time • Universal GMT • Year Month
DisplayLongFormat	Select this check box to display the date and time in long format; for example, April 20, 2009 6:30:00 AM.
Duration	Specifies the length of time the transition takes to complete.
Effect	<p>Click the arrow to expand the property, select the effect you want to apply to the control, and then configure the specific values for the effect. Valid values to select are:</p> <ul style="list-style-type: none"> • Blur <ul style="list-style-type: none"> ◦ Kernel Type ◦ Radius • Drop Shadow

Property	Description
	<ul style="list-style-type: none"> ◦ Color ◦ Direction ◦ Opactiy ◦ Shadow Depth
FlowDirection	<p>Specifies the direction that text and other user interface (UI) elements flow within any parent control. Selections are:</p> <ul style="list-style-type: none"> • Left to Right • Right to Left
Focusable	<p>Select this check box to indicate that the control can receive focus.</p>
FocusVisualStyle	<p>Specifies a property that enables customization of appearance, effects, or other style characteristics that apply to this control when it has focus.</p>
FontFamily	<p>Specifies the font family for the control.</p>
FontSize	<p>Specifies the font size for the control.</p>
FontStretch	<p>Specifies the degree to which a font is condensed or expanded on the screen.</p>
FontStyle	<p>Specifies the font style for the control.</p>
FontWeight	<p>Specifies the weight or thickness of the specified font.</p>
ForceCursor	<p>Select this check box to indicate that the user interface (UI) renders the cursor as declared by the Cursor property.</p>
Foreground	<p>Expand this field to select the foreground color.</p>
GlareBrush	<p>Expand this field to specify the glare effect for the control.</p>
HasError	<p>Select this check box to indicate that validation errors display for any binding.</p>
Height	<p>Specifies the height of the control, in pixels.</p>

Property	Description
HorizontalAlignment	<p>Specifies the horizontal alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
HorizontalContentAlignment	<p>Specifies the horizontal alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
ImageStretch	<p>Specifies how the image should be stretched to fill the destination rectangle. Selections are:</p> <ul style="list-style-type: none"> • Fill • None • Uniform • UniformToFill
InnerBorderBrush	<p>Expand this field to select the color for the inner border.</p>
InnerBorderThickness	<p>Expand this field to select the color for the inner border.</p>
InnerConrnerRadius	<p>Specifies the degree to which the corners of the inner border are rounded.</p>
IsEnabled	<p>Select this check box to indicate that this control is enabled in the user interface (UI).</p>
IsHitTestVisible	<p>Select this check box to indicate that the control can possibly be returned as a hit test result from some portion of its rendered content.</p>

Property	Description
IsManipulationEnabled	Select this check box to indicate that the manipulation events for the control are enabled.
IsSharedSizeScope	Select this check box to indicate that controls are sharing size information.
IsTabStop	Select this check box to indicate that a control is included in tab navigation.
Localization	Click the ellipsis button to open the Localization Configuration dialog box where you can configure the property to localize, the language to use for localization, and the text to localize.
Margin	Specifies the outer margin of a control.
MaxHeight	Specifies the maximum height of a control.
MaxWidth	Specifies the maximum width of a control.
MinHeight	Specifies the minimum height of a control.
MinWidth	Specifies the minimum width of a control.
Name	Specifies the identifying name of the control. The name provides a reference so that code behind, such as event handler code, can refer to a markup element during processing by a XAML processor.
NeedleBrush	Expand this field to select the color for the needle on the outer gauge.
Opacity	Specifies the opacity factor applied to the entire control. Valid values are 0 through 1, with up to 15 decimal places.
OpacityMask	Expand this field to select a color for the opacity mask, which is applied to any alpha-channel masking for the rendered content of this control.
OverridesDefaultStyle	Select this check box to indicate that this control incorporates style properties from theme styles.

Property	Description
Padding	Specifies the amount of padding to apply to the inside edge of each side of a control.
RenderTransformOrigin	<p>Specifies the center point of any possible render transform, relative to the bounds of the element.</p> <ul style="list-style-type: none"> • X is the horizontal origin factor, given as a value between 0 and 1. • Y is the vertical origin factor, given as a value between 0 and 1.
Row	RowBackground
RowSpan	Specifies the total number of rows that child content spans within a grid.
SecondsNeedleBrush	Expand this field to select the color for the second hand.
SelectedDateTime	Specifies the date or date and time that is selected or displayed in the control when in use.
ShowMilliseconds	Select this check box to indicate that milliseconds display on the control.
SnapsToDevicePixels	Select this check box to indicate that rendering for this control uses device-specific pixel settings.
Style	Specifies the style used by this control when it is rendered.
TabIndex	Specifies the order in which controls receive focus when the Tab key is pressed.
Tag	Specifies an arbitrary object value that can be used to store custom information about this control.
Template	Click the ellipsis button to open the Open dialog box to select a template to use.
TickBorderBrush	Expand this field to select the color for the tick border.

Property	Description
TickBrush	Expand this field to select the color for the incremental tick marks.
ToolTip	Specifies the tool-tip that displays for this control in the user interface (UI).
Uid	Specifies the unique identifier (for localization) for this control.
UseLayoutRounding	Specifies a value that determines whether rendering for the object and its visual subtree should use rounding behavior that aligns rendering to whole pixels.
VerticalAlignment	<p>Specifies the vertical alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
VerticalContentAlignment	<p>Specifies the vertical alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
Visibility	<p>Specifies the visibility of this control. Selections are:</p> <ul style="list-style-type: none"> • Collapsed • Hidden • Visible
Width	Specifies the width of the control, in pixels.

Analog Clock



The Analog Clock is a WPF clock control that is designed to simulate a standard clock and automatically runs to the time on your computer in a non-digital format.

See **Date and Time Controls** to view the usage and description of the properties for this control.

Related reference

[Date and Time Controls \(on page 808\)](#)

Date Time Display



The Date Time Display is a WPF clock control that displays date and time using the selected format.

Choose from a short format display (for example, 02/29/08 10:03 A.M.), to the detailed long format (for example, Friday, February 29, 2008 10:03:54 A.M.).

See **Date and Time Controls** to view the usage and description of the properties for this control.

Related reference

[Date and Time Controls \(on page 808\)](#)

Date/Time Picker



The Date Time Picker control is a combination of a calendar and time control that allows you to select the date and time on a display.

See **Date and Time Controls** to view the usage and description of the properties for this control.

Related reference

[Date and Time Controls \(on page 808\)](#)

Duration Picker



The Duration Picker is a WPF control that allows you to add an interactive duration to your display, where you can set the appropriate duration with the up and down arrows, or by selecting the days, hours, minutes, or seconds and entering valid values.

See **Date and Time Controls** to view the usage and description of the properties for this control.

Property	Dual Gauge	Dynamic Gauge	Horizontal Linear Gauge	Parabolic Gauge	Racing Gauge	Radial Gauge	Simple Gauge	Variable Gauge	Vertical Linear Gauge
Background	X	X	X	X	X	X	X	X	X
Border-Brush	X	X	X	X	X	X	X	X	X
Border-Thickness	X	X	X	X	X	X	X	X	X
Caption	X			X	X		X	X	
Caption-Brush	X		X	X	X		X	X	X
Caption-Font-Family	X			X	X		X	X	
Caption-FontSize	X			X	X		X	X	
Caption-Font-Stretch	X			X	X		X	X	
Caption-Font-Style	X			X	X		X	X	
Caption-Font-Weight	X			X	X		X	X	
Center-CapBorderBrush	X			X	X		X	X	

Property	Dual Gauge	Dynamic Gauge	Horizontal Linear Gauge	Parabolic Gauge	Racing Gauge	Radial Gauge	Simple Gauge	Variable Gauge	Vertical Linear Gauge
Center-Cap-Brush	X			X	X		X	X	
ClipTo-Bounds	X	X	X	X	X	X	X	X	X
Color-Threshold1	X								
Color-Threshold2	X								
Color-ThresholdPercentage						X			
Column	X	X	X	X	X	X	X	X	X
Column-Span	X	X	X	X	X	X	X	X	X
Corner-Radius	X		X	X	X		X	X	X
Effect	X	X	X	X	X	X	X	X	X
FillDirection		X							
FillLevel		X							
FlowDirection	X	X	X	X	X	X	X	X	X
FluidBorderBrush			X						X

Property	Dual Gauge	Dynamic Gauge	Horizontal Linear Gauge	Parabolic Gauge	Racing Gauge	Radial Gauge	Simple Gauge	Variable Gauge	Vertical Linear Gauge
Fluid-Brush			X						X
Fluid-Corner-Radius			X						X
Fluid-Opacity-Brush			X						X
Focus-able	X	X	X	X	X	X	X	X	X
FocusVisualStyle	X	X	X	X	X	X	X	X	X
Font-Family	X	X	X	X	X	X	X	X	X
FontSize	X	X	X	X	X	X	X	X	X
FontStretch	X	X	X	X	X	X	X	X	X
FontStyle	X	X	X	X	X	X	X	X	X
FontWeight	X	X	X	X	X	X	X	X	X
Force-Cursor	X	X	X	X	X	X	X	X	X
Fore-ground	X	X	X	X	X	X	X	X	X
Glare-Brush	X		X	X	X		X	X	X
Height	X	X	X	X	X	X	X	X	X
High-Color-		X							

Property	Dual Gauge	Dynamic Gauge	Horizontal Linear Gauge	Parabolic Gauge	Racing Gauge	Radial Gauge	Simple Gauge	Variable Gauge	Vertical Linear Gauge
IsEnabled	X	X	X	X	X	X	X	X	X
IsFillReversed						X			
IsHitTestVisible	X	X	X	X	X	X	X	X	X
IsManipulationEnabled	X	X	X	X	X	X	X	X	X
IsNumbersVisible			X						X
IsRunning		X				X			
IsSharedSizeScope	X	X	X	X	X	X	X	X	X
IsTabStop	X	X	X	X	X	X	X	X	X
Localization	X	X	X	X	X	X	X	X	X
LowColorThreshold1		X							
LowColor-		X							

Property	Dual Gauge	Dynamic Gauge	Horizontal Linear Gauge	Parabolic Gauge	Racing Gauge	Radial Gauge	Simple Gauge	Variable Gauge	Vertical Linear Gauge
Threshold2									
LowFill		X				X			
Low-Range-Brush								X	
Margin	X	X	X	X	X	X	X	X	X
Max-Height	X	X	X	X	X	X	X	X	X
Maximum	X	X	X	X	X	X	X	X	X
MaxWidth	X	X	X	X	X	X	X	X	X
Medium-Fill		X				X			
Meter-Background-Brush					X				
Meter-Border-Brush					X				
Meter-Font-Family					X				
Meter-FontSize					X				
Meter-Font-Stretch					X				

Property	Dual Gauge	Dynamic Gauge	Horizontal Linear Gauge	Parabolic Gauge	Racing Gauge	Radial Gauge	Simple Gauge	Variable Gauge	Vertical Linear Gauge
Default-Style									
Padding	X	X	X	X	X	X	X	X	X
Render-TransformOrigin	X	X	X	X	X	X	X	X	X
Row	X			X	X	X	X	X	X
RowSpan	X			X	X	X	X	X	X
SnapsToDevicePixels	X	X	X	X	X	X	X	X	X
StrokeThickness	X	X		X	X		X	X	X
Style	X	X	X	X	X	X	X	X	X
TabIndex	X	X	X	X	X	X	X	X	X
Tag	X	X	X	X	X	X	X	X	X
Template	X	X	X	X	X	X	X	X	X
TickBrush	X		X	X	X		X	X	X
TickFrequency			X						

Property	Dual Gauge	Dynamic Gauge	Horizontal Linear Gauge	Parabolic Gauge	Racing Gauge	Radial Gauge	Simple Gauge	Variable Gauge	Vertical Linear Gauge
Tick-Placement			X						
Tick-Ring-Brush	X			X	X		X		
Ticks	X		X	X	X		X	X	X
Title-FontSize						X			
TitleText						X			
Title-TextColor						X			
ToolTip	X	X	X	X	X	X	X	X	X
Uid	X	X	X	X	X	X	X	X	X
Unit-FontSize						X			
UnitText						X			
UnitText-Color						X			
Use-Layout-Rounding	X	X	X	X	X	X	X	X	X
Value	X	X	X	X	X	X	X	X	X
Value-Back-		X							

Property	Dual Gauge	Dynamic Gauge	Horizontal Linear Gauge	Parabolic Gauge	Racing Gauge	Radial Gauge	Simple Gauge	Variable Gauge	Vertical Linear Gauge
ground-Color									
Value-FontSize						X			
Value-Format		X							
Value-String-Format						X			
Vertical-Alignment	X	X	X	X	X	X	X	X	X
Vertical-Content-Alignment	X	X	X	X	X	X	X	X	X
Visibility	X	X	X	X	X	X	X	X	X
Width	X	X	X	X	X	X	X	X	X

Gauge Controls Property Descriptions

The following table shows the properties descriptions used by all control:

Property	Description
AllowDrop	Select this check box to indicate that the control can accept data that is dragged onto it.
Background	Expand this field to select the background color of the selected control.
BorderBrush	Expand this field to select the border color of the selected control.

Property	Description
BorderThickness	Specifies the border thickness of a control, in pixels.
Caption	Specifies the caption for the main control.
CaptionBrush	Expand this field to select the color for the main gauge caption.
CaptionFontFamily	Specifies the font family for the control's caption.
CaptionFontSize	Specifies the font size for the control's caption.
CaptionFontStretch	Specifies the degree to which a caption's font is condensed or expanded on the screen.
CaptionFontStyle	Specifies the font style for the control's caption.
CaptionFontWeight	Specifies the font weight for the control's caption.
CenterCapBorderBrush	Expand this field to select a color for the outside edge of the center cap.
CenterCapBrush	Expand this field to select a color for the center cap.
ClipToBounds	Select this check box to clip the content of this control (or content coming from the child elements of this control) to fit into the size of the containing control.
ColorThreshold1	Specifies the percentage between the minimum and maximum where the control will change to the medium fill.
ColorThreshold2	Specifies the percentage between the minimum and maximum where the control will change to the high fill.
ColorThresholdPercentage	Specifies the percentage between the minimum and maximum where the control will change to the medium fill.
Column	Specifies the column within the grid where the control is located.

Property	Description
ColumnSpan	Specifies the total number of columns that child content spans within a grid.
CornerRadius	Specifies the degree to which the corners of a control are rounded.
Effect	<p>Click the arrow to expand the property, select the effect you want to apply to the control, and then configure the specific values for the effect. Valid values to select are:</p> <ul style="list-style-type: none"> • Blur <ul style="list-style-type: none"> ◦ Kernel Type ◦ Radius • Drop Shadow <ul style="list-style-type: none"> ◦ Color ◦ Direction ◦ Opactiy ◦ Shadow Depth
FillDirection	<p>Specifies the direction of the fill. Selections are:</p> <ul style="list-style-type: none"> • From Left • From Right • From Center
FillLevel	Specifies the level of fill based on the values of the Maximum, MidValue, and Minimum properties.
FlowDirection	<p>Specifies the direction that text and other user interface (UI) elements flow within any parent control. Selections are:</p> <ul style="list-style-type: none"> • Left to Right • Right to Left
FluidBorderBrush	Expand this field to select the border color for the fluid content of the selected control.

Property	Description
FluidBrush	Expand this field to select the color for the fluid portion of the selected control.
FluidCornerRadius	Specifies the degree to which the corners of the fluid content a control are rounded.
FluidOpacityBrush	Expand this field to select the color for the opacity level of the fluid.
Focusable	Select this check box to indicate that the control can receive focus.
FocusVisualStyle	Specifies a property that enables customization of appearance, effects, or other style characteristics that apply to this control when it has focus.
FontFamily	Specifies the font family for the control.
FontSize	Specifies the font size for the control.
FontStretch	Specifies the degree to which a font is condensed or expanded on the screen.
FontStyle	Specifies the font style for the control.
FontWeight	Specifies the weight or thickness of the specified font.
ForceCursor	Select this check box to indicate that the user interface (UI) renders the cursor as declared by the Cursor property.
Foreground	Expand this field to select the foreground color.
GlareBrush	Expand this field to specify the glare effect for the control.
Height	Specifies the height of the control, in pixels.
HighColorThreshold1	Specifies the percentage of the maximum where the control will change to the medium fill.
HighColorThreshold2	Specifies the percentage of the maximum where the control will chane to the high fill.

Property	Description
HighFill	Specifies the brush that describes the high fill of the control.
HighRangeBrush	Expand this field to select the color for the high range display.
HorizontalAlignment	<p>Specifies the horizontal alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
HorizontalContentAlignment	<p>Specifies the horizontal alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
ImageStretch	<p>Specifies how the image should be stretched to fill the destination rectangle. Selections are:</p> <ul style="list-style-type: none"> • Fill • None • Uniform • UniformToFill
InnerBackground	Expand this field to select the background color for the fluid indicator; that is, the non-fluid portion of the control.
InnerBorderBrush	Expand this field to select the color for the inner border.
InnerBorderThickness	Expand this field to select the color for the inner border.
InnerCaption	Specifies the caption for the inner gauge.

Property	Description
InnerCaptionBrush	Expand this field to select the color for the caption for the inner gauge.
InnerCaptionFontFamily	Specifies the font family for the caption for the inner gauge.
InnerCaptionFontSize	Specifies the font size for the caption for the inner gauge.
InnerCaptionFontStretch	Specifies the degree to which the font for caption for the inner gauge is condensed or expanded on the screen.
InnerCaptionFontWeight	Specifies the font weight or thickness for the caption for the inner gauge.
InnerCornerRadius	Specifies the degree to which the corners of the inner border are rounded.
InnerFontFamily	Specifies the font family for the inner gauge.
InnerFontSize	Specifies the font size for the inner gauge.
InnerFontStretch	Specifies the degree to which the font for the inner gauge is condensed or expanded on the screen.
InnerFontStyle	Specifies the font style for the inner gauge.
InnerFontWeight	Specifies the weight or thickness of the font for the inner gauge.
InnerForeground	Expand this field to select the foreground color for the inner gauge.
InnerMaximum	Specifies the maximum value of the range for the inner gauge.
InnerMinimum	Specifies the minimum value of the range for the inner gauge.
InnerNeedleBrush	Expand this field to select the color for the needle on the inner gauge.
InnerNeedleThickness	Specifies the thickness of the outside edge of the needle on the inner gauge.

Property	Description
InnerTickBrush	Expand this field to select the color for the tick marks on the inner gauge.
InnerTickRingBrush	Expand this field to select the color for the tick ring of the inner gauge.
InnerTicks	Specifies the values to display.
InnerValue	Specifies the current magnitude of the range control for the inner gauge.
IsEnabled	Select this check box to indicate that this control is enabled in the user interface (UI).
IsFillReversed	Specifies whether the gauge fills from left or right. Select this check box to indicate that the gauge will fill from right to left. Clear this check box to indicate that the gauge will fill from left to right.
IsHitTestVisible	Select this check box to indicate that the control can possibly be returned as a hit test result from some portion of its rendered content.
IsManipulationEnabled	Select this check box to indicate that the manipulation events for the control are enabled.
IsNumbersVisible	Select this check box to indicate that the numbers corresponding to the tick marks are visible.
IsRunning	Specifies whether the stopwatch timer is running on the control.
IsSharedSizeScope	Select this check box to indicate that controls are sharing size information.
IsTabStop	Select this check box to indicate that a control is included in tab navigation.
Localization	Click the ellipsis button to open the Localization Configuration dialog box where you can configure the property to localize, the language to use for localization, and the text to localize.

Property	Description
LowColorThreshold1	Specifies the percentage of the minimum where the control will change to the medium fill.
LowColorThreshold2	Specifies the percentage of the minimum where the control will change to the low fill.
LowFill	Specifies the brush that describes the low fill of the control.
LowRangeBrush	Expand this field to select the color for the low range display.
Margin	Specifies the outer margin of a control.
MaxHeight	Specifies the maximum height of a control.
Maximum	Specifies the maximum value of the range of the control.
MaxWidth	Specifies the maximum width of a control.
MediumFill	Specifies the brush that describes the medium fill of the control.
MeterBackgroundBrush	Expand this field to select a color for the meter background.
MeterBorderBrush	Expand this field to select a color for the meter border.
MeterFontFamily	Specifies the font family for the meter value.
MeterFontSize	Specifies the font size for the meter value.
MeterFontStretch	Specifies the degree to which the meter value font is condensed or expanded on the screen.
MeterFontStyle	Specifies the font style for the meter value.
MeterFontWeight	Specifies the weight or thickness of the specified font for the meter value.
MeterForeground	Expand this field to select the color for the value that appears in the meter area.

Property	Description
MidRangeBrush	Expand this field to select the color for the mid range display.
MinHeight	Specifies the minimum height of a control.
Minimum	Specifies the minimum value of the range of the control.
MinWidth	Specifies the minimum width of a control.
Name	Specifies the identifying name of the control. The name provides a reference so that code behind, such as event handler code, can refer to a markup element during processing by a XAML processor.
NeedleBrush	Expand this field to select the color for the needle on the outer gauge.
NeedleThickness	Specifies the thickness of the outside edge of the needle on the outer gauge.
Opacity	Specifies the opacity factor applied to the entire control. Valid values are 0 through 1, with up to 15 decimal places.
OpacityMask	Expand this field to select a color for the opacity mask, which is applied to any alpha-channel masking for the rendered content of this control.
OverridesDefaultStyle	Select this check box to indicate that this control incorporates style properties from theme styles.
Padding	Specifies the amount of padding to apply to the inside edge of each side of a control.
RenderTransformOrigin	<p>Specifies the center point of any possible render transform, relative to the bounds of the element.</p> <ul style="list-style-type: none"> • X is the horizontal origin factor, given as a value between 0 and 1. • Y is the vertical origin factor, given as a value between 0 and 1.

Property	Description
Row	RowBackground
RowSpan	Specifies the total number of rows that child content spans within a grid.
SnapsToDevicePixels	Select this check box to indicate that rendering for this control uses device-specific pixel settings.
StrokeThickness	Specifies the width of the control's outline.
Style	Specifies the style used by this control when it is rendered.
TabIndex	Specifies the order in which controls receive focus when the Tab key is pressed.
Tag	Specifies an arbitrary object value that can be used to store custom information about this control.
Template	Click the ellipsis button to open the Open dialog box to select a template to use.
TickBrush	Expand this field to select the color for the incremental tick marks.
TickFrequency	Specifies the interval between tick marks.
TickPlacement	<p>Specifies the position of tick marks with respect to the track of the Slider. Selections are:</p> <ul style="list-style-type: none"> • None • Both • TopLeft • BottomRight
TickRingBrush	Expand this field to select the color for the tick ring of the outer gauge.
Ticks	For linear gauges and sliders, click the ellipsis button to open the Tick Property Editor dialog box to enter the marks to display for a Slider.
TitleFontSize	Specifies the font size for the title.

Property	Description
TitleText	Specifies the title of the control.
TitleTextColor	Specifies the brush that colors the title of the control.
ToolTip	Specifies the tool-tip that displays for this control in the user interface (UI).
Uid	Specifies the unique identifier (for localization) for this control.
UnitFontSize	Specifies the font size for the units display values.
UnitText	Specifies the unit text for the control.
UnitTextColor	Specifies the brush that colors the unit text of the control.
UseLayoutRounding	Specifies a value that determines whether rendering for the object and its visual subtree should use rounding behavior that aligns rendering to whole pixels.
Value	Specifies the current magnitude of the range control.
ValueBackgroundColor	Expand this field to select the background color of the values for the selected control.
ValueFontSize	Specifies the font size of the value for the control.
ValueFormat	Specifies the format the value displays in.
ValueStringFormat	Specifies the format string to use when converting the Value property to a string.
VerticalAlignment	<p>Specifies the vertical alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch

Property	Description
VerticalContentAlignment	Specifies the vertical alignment of the control's content. Selections are: <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
Visibility	Specifies the visibility of this control. Selections are: <ul style="list-style-type: none"> • Collapsed • Hidden • Visible
Width	Specifies the width of the control, in pixels.

Dual Gauge



A Dual Gauge control is a radial gauge with a circular shape. The dual gauge has the ability to display two different variables from within the same display face.

The gauge contains two needles, two sets of ticks and two tick rings. The default value for the control is the Value property, and operators may also want to find to the InnerValue property.

The Batch Generic Button control causes multiple events to occur each time the button is pressed.

See **Gauge Controls** to view the usage and description of the properties for this control.

Related reference

[Gauge Controls \(on page 820\)](#)

Dynamic Gauge



The Dynamic Gauge is a WPF control that is a type of linear gauge with the ability to display values from the left, right, and center.

The Dynamic Gauge has two threshold properties that allow you to select where the fill will change color.

The Batch Generic Button control causes multiple events to occur each time the button is pressed.

See **Gauge Controls** to view the usage and description of the properties for this control.

Related reference

[Gauge Controls \(on page 820\)](#)

Horizontal Linear Gauge



The Horizontal Linear Gauge control displays a horizontally oriented, customizable metered numeric output.

See **Gauge Controls** to view the usage and description of the properties for this control.

Related reference

[Gauge Controls \(on page 820\)](#)

Parabolic Gauge



A Parabolic Gauge control is a radial gauge that measures a set numeric amount within a framed half circle display area.

The parabolic gauge display only one set of values within the display face.

See **Gauge Controls** to view the usage and description of the properties for this control.

Related reference

[Gauge Controls \(on page 820\)](#)

Racing Gauge



A Racing Gauge control is a radial gauge that displays a set amount in two different formats.

This gauge contains a meter-style numerical display, as well as one needle, one set of ticks, and a tick ring.

See **Gauge Controls** to view the usage and description of the properties for this control.

Related reference

[Gauge Controls \(on page 820\)](#)

Radial Gauge



A Radial Gauge is a WPF control with the ability to display one value in two different ways from within the same display face.

The Radial Gauge control displays the numeric value in the center of the control and the non-numeric value in the rings around the outside.

See **Gauge Controls** to view the usage and description of the properties for this control.

Related reference

[Gauge Controls \(on page 820\)](#)

Simple Gauge



A Simple Gauge control is a radial gauge that displays data as a set amount within a half-circle unframed display.

See **Gauge Controls** to view the usage and description of the properties for this control.

Related reference

[Gauge Controls \(on page 820\)](#)

Variable Gauge



A Variable Gauge control is a radial gauge that displays data in three user-defined range variables. This gauge can be used to display numeric data on a non-numeric graphic scale.

See **Gauge Controls** to view the usage and description of the properties for this control.

Related reference

[Gauge Controls \(on page 820\)](#)

Vertical Linear Gauge



The Vertical Linear Gauge control displays a vertically oriented, customizable metered numeric output.

See **Gauge Controls** to view the usage and description of the properties for this control.

Related reference[Gauge Controls \(on page 820\)](#)

Grids and Panels

Workflow provides a series of grids and panels to use in forms and user display authoring.

Grids are used to precisely position content in rows and columns. Panels are components that control the rendering of other controls; for example, their size and dimensions, their positions, and the arrangement of their child content.

The following topics describe the controls available in the Grids/Panels list.

Grids and Panels Controls Property Usage

The following table shows the properties used by each control:

Property	Border	Canvas	Data Grid	Dock Panel	Grid	Grid Splitter	Ink Presenter	Stack Panel	Uniform Grid	View Box	View-port 3D	Wrap Panel
Allow-Drop	X	X	X	X	X	X	X	X	X	X	X	X
AlternatingRow-Background			X									
Alternation-Count			X									
Automatic					X							
Background	X	X	X	X	X	X	X	X	X			X

Property	Border	Canvas	Data Grid	Dock Panel	Grid	Grid Splitter	Ink Presenter	Stack Panel	Uniform Grid	View Box	View-port 3D	Wrap Panel
Border-Brush	X		X			X						
Border-Thickness	X		X			X						
CacheMode							X					
Clip							X					
Clip-To-Bounds	X	X	X	X	X	X		X	X	X	X	X
Column	X	X	X	X	X	X	X	X	X	X	X	X
Columns									X			
ColumnSpan	X	X		X	X	X	X	X	X	X	X	X
Column-Header-Height			X									
Command			X									
CornerRadius	X											
Cursor							X					

Property	Border	Canvas	Data Grid	Dock Panel	Grid	Grid Splitter	Ink Presenter	Stack Panel	Uniform Grid	View Box	View-port 3D	Wrap Panel
Data-Context							X					
Drag-Increment						X						
Effect	X	X	X	X	X	X	X	X	X	X	X	X
Flow-Direction	X	X	X	X	X	X	X	X	X	X	X	X
Fo-cus-able	X	X	X	X	X	X		X	X	X	X	X
Fo-cusVi-sual-Style	X	X	X	X	X	X		X	X	X	X	X
Font-Family			X			X						
FontSize			X			X						
FontStretch			X			X						
FontStyle			X			X						
FontWeight			X			X						
Force-Cursor	X	X	X	X	X	X		X	X	X	X	X
Fore-ground			X			X						

Property	Border	Canvas	Data Grid	Dock Panel	Grid	Grid Splitter	Ink Presenter	Stack Panel	Uniform Grid	View Box	View-port 3D	Wrap Panel
Height	X	X	X	X	X	X	X	X	X	X	X	X
Hidden-Columns			X									
Horizontal-Alignment	X	X	X	X	X	X	X	X	X	X	X	X
Horizontal-Content-Alignment			X			X						
Horizontal-Grid-Lines-Brush			X									
Input-Table			X									
IsEnabled	X	X	X	X	X	X		X	X	X	X	X
IsHit-Test-Visible	X	X	X	X	X	X	X	X	X	X	X	X
IsManipu-	X	X	X	X	X	X		X	X	X	X	X

Property	Border	Canvas	Data Grid	Dock Panel	Grid	Grid Splitter	Ink Presenter	Stack Panel	Uniform Grid	View Box	View-port 3D	Wrap Panel
Max-Height	X	X	X	X	X	X	X	X	X	X	X	X
MaxWidth	X	X	X	X	X	X	X	X	X	X	X	X
Min-Column-Width			X									
Min-Height	X	X	X	X	X	X	X	X	X	X	X	X
Min-Width	X	X	X	X	X	X	X	X	X	X	X	X
Name	X	X	X	X	X	X	X	X	X	X	X	X
Opacity	X	X	X	X	X	X	X	X	X	X	X	X
Opacity-Mask	X	X	X	X	X	X	X	X	X	X	X	X
Orientation								X				X
Overrides-Default-Style	X	X	X	X	X	X		X	X	X	X	X
Padding	X		X			X						
Pixel					X							
Pre-view-Style						X						

Property	Border	Canvas	Data Grid	Dock Panel	Grid	Grid Splitter	Ink Presenter	Stack Panel	Uniform Grid	View Box	View-port 3D	Wrap Panel
Template			X			X						
ToolTip	X	X	X	X	X	X	X	X	X	X	X	X
Uid	X	X	X	X	X	X		X	X	X	X	X
Use-Layout-Rounding	X	X	X	X	X	X	X	X	X	X	X	X
Vertical-Alignment	X	X	X	X	X	X	X	X	X	X	X	X
Vertical-Content-Alignment			X			X						
Vertical-Grid-Lines-Brush			X									
Visibility	X	X	X	X	X	X	X	X	X	X	X	X
Width	X	X	X	X	X	X	X	X	X	X	X	X

Grids and Panels Controls Property Descriptions

The following table shows the properties descriptions used by all control:

Property	Description
AllowDrop	Select this check box to indicate that the control can accept data that is dragged onto it.
AlternatingRowBackground	Expand this field to select the background color to display on alternating rows.
AlternationCount	Specifies the number of alternating item containers or rows to display.
Automatic	<p>The selected row or column automatically resizes to the width or height of the cell's largest object.</p> <div data-bbox="820 709 1422 1066" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: This setting should be used only after objects are placed within the grid cells. This allows the row or column to properly resize with its child objects. If a row or column is empty and set to automatic sizing, it collapses.</p> </div>
ArrowBackground	Expand this field to select the background color for an arrow button control.
ArrowBorderBrush	Expand this field to select the border color of the selected arrow control.
ArrowDesign	<p>Specifies the appearance of the arrow. Available selections are:</p> <ul style="list-style-type: none"> • Pointer • Barbed • Triangle
ArrowDirection	<p>Specifies the direction of the arrow. Available selections are:</p> <ul style="list-style-type: none"> • Up • Down • Right • Left

Property	Description
ArrowPadding	Specifies the horizontal and vertical alignment of the arrow within the button.
Background	Expand this field to select the background color of the selected control.
BorderBrush	Expand this field to select the border color of the selected control.
BorderThickness	Specifies the border thickness of a control, in pixels.
CacheMode	Specifies a value that indicates that rendered content should be cached when possible.
CalendarButtonStyle	Specifies the style of the calendar's buttons.
Clip	<p>Specifies the geometric shape used to define the outline of the image. Selections are:</p> <ul style="list-style-type: none"> • Hover • Press • Release
ClipToBounds	Select this check box to clip the content of this control (or content coming from the child elements of this control) to fit into the size of the containing control.
Column	Specifies the column within the grid where the control is located.
Columns	Specifies the number of columns in the grid.
ColumnSpan	Specifies the total number of columns that child content spans within a grid.
ColumnHeaderHeight	Specifies the height of the column header row, in pixels. The default is 20 pixels.
Command	Specifies the command to invoke when this control is pressed or clicked.

Property	Description
CornerRadius	Specifies the degree to which the corners of a control are rounded.
Cursor	Specifies the cursor image to display when the mouse pointer hovers over an object.
DataContext	Specifies the data context for a control when it participates in data binding.
DragIncrement	Specifies the minimum distance that you must drag a mouse to resize rows or columns.
Effect	<p>Click the arrow to expand the property, select the effect you want to apply to the control, and then configure the specific values for the effect. Valid values to select are:</p> <ul style="list-style-type: none"> • Blur <ul style="list-style-type: none"> ◦ Kernel Type ◦ Radius • Drop Shadow <ul style="list-style-type: none"> ◦ Color ◦ Direction ◦ Opacity ◦ Shadow Depth
FlowDirection	<p>Specifies the direction that text and other user interface (UI) elements flow within any parent control. Selections are:</p> <ul style="list-style-type: none"> • Left to Right • Right to Left
Focusable	Select this check box to indicate that the control can receive focus.
FocusVisualStyle	Specifies a property that enables customization of appearance, effects, or other style characteristics that apply to this control when it has focus.
FontFamily	Specifies the font family for the control.

Property	Description
FontSize	Specifies the font size for the control.
FontStretch	Specifies the degree to which a font is condensed or expanded on the screen.
FontStyle	Specifies the font style for the control.
FontWeight	Specifies the weight or thickness of the specified font.
ForceCursor	Select this check box to indicate that the user interface (UI) renders the cursor as declared by the Cursor property.
Foreground	Expand this field to select the foreground color.
GridColumnms	Specifies the columns for the data grid. The columns are chosen from a list of attributes and specifications derived from the available quality variables.
GridRows	Specifies the rows for the data grid. The rows are chosen from a list of available quality variables.
GridHeadersVisible	Select this check box to display the header row in the grid.
GridLinesVisible	Select this check box to display the grid lines.
HeaderBackground	Expand this field to select the header background color.
HeaderFontFamily	Specifies the font family for the text in the header.
HeaderFontSize	Specifies the font size for the text in the header.
HeaderFontWeight	Specifies the font weight for the text in the header.
HeaderForeground	Expand this field to select the header text color.
Height	Specifies the height of the control, in pixels.
HiddenColumns	Specifies the column(s) to omit from the displayed grid, separated by a comma. Column numbers start

Property	Description
	at zero (0). For example, to hide the first, fourth, and sixth columns, enter 0, 3, 5.
HorizontalAlignment	<p>Specifies the horizontal alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
HorizontalContentAlignment	<p>Specifies the horizontal alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
HorizontalGridLinesBrush	Expand this field to select the color for the horizontal grid lines. The default is black.
InputTable	Specifies the table to display in the grid. The InputTable can be defined in a number of ways: in code behind, configured from a data item, or passed in from a workflow (that is, either created in the workflow itself or passed in from an Historian read, which return a DataTable).
IsEnabled	Select this check box to indicate that this control is enabled in the user interface (UI).
IsHitTestVisible	Select this check box to indicate that the control can possibly be returned as a hit test result from some portion of its rendered content.
IsManipulationEnabled	Select this check box to indicate that the manipulation events for the control are enabled.
IsSharedSizeScope	Select this check box to indicate that controls are sharing size information.

Property	Description
IsTabStop	Select this check box to indicate that a control is included in tab navigation.
ItemHeight	Specifies the height of all items contained within a Wrap Panel.
ItemWidth	Specifies the width of all items contained within a Wrap Panel.
KeyboardIncrement	Specifies the distance that each press of an arrow key moves a Grid Splitter control.
Language	Specifies the localization language that applies to an object.
LastChildFill	Select this check box to indicate that the last child control within a Dock Panel stretches to fill the remaining available space.
Localization	Click the ellipsis button to open the Localization Configuration dialog box where you can configure the property to localize, the language to use for localization, and the text to localize.
Margin	Specifies the outer margin of a control.
MaxColumnWidth	Specifies the maximum width of the columns and headers in the grid.
MaxHeight	Specifies the maximum height of a control.
MaxWidth	Specifies the maximum width of a control.
MinColumnWidth	Specifies the minimum width of the columns and headers in the grid.
MinHeight	Specifies the minimum height of a control.
MinWidth	Specifies the minimum width of a control.
Name	Specifies the identifying name of the control. The name provides a reference so that code behind, such as event handler code, can refer to a markup element during processing by a XAML processor.

Property	Description
Opacity	Specifies the opacity factor applied to the entire control. Valid values are 0 through 1, with up to 15 decimal places.
OpacityMask	Expand this field to select a color for the opacity mask, which is applied to any alpha-channel masking for the rendered content of this control.
Orientation	<p>Specifies the orientation of a control. Selections are:</p> <ul style="list-style-type: none"> • Portrait • Landscape
OverridesDefaultStyle	Select this check box to indicate that this control incorporates style properties from theme styles.
Padding	Specifies the amount of padding to apply to the inside edge of each side of a control.
Pixel	The selected row or column in the grid becomes a fixed size and remains fixed even if the grid is resized.
PreviewStyle	Specifies the style that customizes the appearance, effects, or other style characteristics for the Grid Splitter control preview indicator that displays when the ShowsPreview property is set to True.
Projection	Specifies the 3-D effect to apply to the current object.
RenderTransform	Specifies the transform information that affects the rendering position of an object.
RenderTransformOrigin	Specifies the center point of any possible render transform, relative to the bounds of the element.

Property	Description
	<ul style="list-style-type: none"> • X is the horizontal origin factor, given as a value between 0 and 1. • Y is the vertical origin factor, given as a value between 0 and 1.
ResizeBehavior	<p>Specifies the columns or rows that are resized relative to the column or row the GridSplitter control is defined for. Valid selections are:</p> <ul style="list-style-type: none"> • Based On Alignment • Current and Next • Previous and Current • Previous and Next
ResizeDirection	<p>Specifies whether the GridSplitter control resizes rows or columns. Valid selections are:</p> <ul style="list-style-type: none"> • Auto • Columns • Rows
Row	Specifies the row background.
Rows	Specifies the number of rows in the grid.
RowBackground	Expand this field to select the background color for the row.
RowHeaderWidth	Specifies the width of the row header column.
RowHeight	Specifies the default height for all rows.
RowSpan	Specifies the total number of rows that child content spans within a grid.
SelectedRows	Specifies the selected row(s) as a new table.
ShowGridLines	Select this check box to indicate that grid lines are visible within this grid.
ShowsPreview	Select this check box to indicate that the Grid Splitter control updates the column or row size as the control is dragged.

Property	Description
SnapsToDevicePixels	Select this check box to indicate that rendering for this control uses device-specific pixel settings.
Star (relatonal)	<p>A row or column height or width set to relational sizing becomes a proportional size.</p> <p>Star values are ratios calculated from the remaining available space in the grid and are subject to a certain hierarchy. The total available space in a grid is allocated first to rows and columns with fixed pixel row height / column width. Then, space is allocated to rows and columns that are set to automatically size. Finally, rows and columns with relational heights and widths share a percentage of the grid's remaining space.</p> <p>For example, consider a grid with two columns that is 1000 pixels in width. One column has a fixed size of 100 pixels; the other column is set to relational sizing.</p> <p>In order to determine the available space for the relational column, subtract the width of the column set in pixels from the total width of the grid (in pixels): $1000 - 100 = 900$ pixels of remaining space.</p> <p>Since the default star value is 1, it is easy to calculate the number of pixels per star in this example. Pixels per star are calculated by dividing the total available width (in pixels) by the total number of star values in the grid: $900 \div 1 = 900$ pixels per star.</p> <p>Therefore, the size of the relational column is 900 pixels (900 pixels x 1 star).</p> <p>Now consider if a third column is added with a star value of 2. The grid still has 900 pixels of remaining available space, but now the star values total 3</p>

Property	Description
	<p>(1 star from the first relational column, 2 stars from the new column): $900 \div 3 = 300$ pixels per star.</p> <p>Therefore the first relational column is now a size of 300 pixels (300 pixels x 1 star) and the new column is a size of 600 pixels (300 pixels x 2 stars).</p> <p>The grid's child objects inherit a number of attached dependency properties. The Row and Column properties specify the starting row and column that displays the object. RowSpan and Column Span show the number of cells the object spans, either vertically or horizontally. And, the Margin property specifies the amount of unused space, in pixels, between the grid cell lines and the object.</p>
Stretch	<p>Specifies how an Image stretches to fill the destination rectangle. Selections are:</p> <ul style="list-style-type: none"> • Fill • None (does not apply to Scale Box) • Uniform (does not apply to Scale Box) • UniformFill (does not apply to Scale Box)
StretchDirection	<p>Specifies how the image is scaled. Selections are:</p> <ul style="list-style-type: none"> • Vertical • Horizontal
Strokes	<p>Specifies the strokes that the control displays.</p>
Style	<p>Specifies the style used by this control when it is rendered.</p>
TabIndex	<p>Specifies the order in which controls receive focus when the Tab key is pressed.</p>
Tag	<p>Specifies an arbitrary object value that can be used to store custom information about this control.</p>

Property	Description
Template	Click the ellipsis button to open the Open dialog box to select a template to use.
ToolTip	Specifies the tool-tip that displays for this control in the user interface (UI).
Uid	Specifies the unique identifier (for localization) for this control.
UseLayoutRounding	Specifies a value that determines whether rendering for the object and its visual subtree should use rounding behavior that aligns rendering to whole pixels.
VerticalAlignment	<p>Specifies the vertical alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
VerticalContentAlignment	<p>Specifies the vertical alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
VerticalGridLinesBrush	Specifies the color used to draw the vertical grid lines. The default is black.
Visibility	<p>Specifies the visibility of this control. Selections are:</p> <ul style="list-style-type: none"> • Collapsed • Hidden • Visible
Width	Specifies the width of the control, in pixels.

Add a child object to a border

The Border control acts as a base for other layout objects that do not have border and border thickness properties, such as a canvas or a grid.

Before you begin

**Note:**

You can add only one container to a border. If you drag another container to the border and add it as a child, the previous child container is overwritten.

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. Expand the control group you want to work with.
6. Click and drag the control to the desired position in the **Designer**.

**Note:**

To apply the properties of a control to another control, click and drag the control to another control in the document tree. The Applying **<control (control name)>** to **Target** dialog box appears where you can select the option you want to use to apply to the control.

7. From the **Grids/Panels** list, click and drag a container to the document tree and drop it on the border control.
The Applying **<container name>** to Target Border dialog box appears, where **<container name>** is the type of container you are adding.
8. Select the **As a Child of Border** option, and then click **OK**.
9. Configure the properties, as required.

**Note:**

To view the full list of properties, select the **Advanced** check box.

10. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Add an object to a dock panel

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. Expand the control group you want to work with.
6. Click and drag the control to the desired position in the **Designer**.



Note:

To apply the properties of a control to another control, click and drag the control to another control in the document tree. The Applying **<control (control name)> to Target** dialog box appears where you can select the option you want to use to apply to the control.

7. Right-click the dock panel and select **Edit Layer**.
The dock panel's editor is initiated, which is indicated by an orange border.
8. From the **Toolbox**, click the object you want to add to the dock panel.
9. Move the mouse pointer over the dock panel's editable area.
The mouse cursor changes to a bold plus sign with the word "object."
10. Click in the dock panel to add the object.



Important:

Objects cannot be explicitly positioned in a dock panel; therefore, objects cannot be drawn onto the dock panel and/or selected and moved.

11. Configure the properties, as required.



Note:

To view the full list of properties, select the **Advanced** check box.

12. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Add rows and columns to a grid

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. Expand the control group you want to work with.
6. Click and drag the control to the desired position in the **Designer**.



Note:

To apply the properties of a control to another control, click and drag the control to another control in the document tree. The Applying **<control (control name)> to Target** dialog box appears where you can select the option you want to use to apply to the control.

7. Right-click the grid and select **Edit Layer**.
The grid's editor is initiated, which is indicated by an orange border. By default, there is one row (Row0) and one column (Column0) in the grid, and the row and column edit buttons are visible.
8. Click **Insert new row**.
A new row (Row1) appears with its own set of buttons.
9. Click **Insert new column**.
A new column (Column1) appears with its own set of buttons.
10. To add a row between existing rows, click **Insert new row** in the row above where you want to add the row. For example, if you want to add a new row between Row4 and Row5, click **Insert new row** in Row4.
11. To add a column between existing columns, click **Insert new column** in the column to the left of where you want to add the new column. For example, if you want to add a new column between Column4 and Column5, click **Insert new column** in Column 4.
12. To remove a row or column, click **Delete Row** or **Delete Column** for the specific row or column.
13. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Size rows and columns for a grid

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. Expand the control group you want to work with.
6. Click and drag the control to the desired position in the **Designer**.



Note:

To apply the properties of a control to another control, click and drag the control to another control in the document tree. The Applying **<control (control name)>** to **Target** dialog box appears where you can select the option you want to use to apply to the control.

7. Right-click the grid and select **Edit Layer**.
The grid's editor is initiated, which is indicated by an orange border. By default, there is one row (Row0) and one column (Column0) in the grid, and the row and column edit buttons are visible.
8. Click **Insert new row**.
A new row (Row1) appears with its own set of buttons.
9. Click **Insert new column**.
A new column (Column1) appears with its own set of buttons.
10. To add a row between existing rows, click **Insert new row** in the row above where you want to add the row. For example, if you want to add a new row between Row4 and Row5, click **Insert new row** in Row4.
11. To add a column between existing columns, click **Insert new column** in the column to the left of where you want to add the new column. For example, if you want to add a new column between Column4 and Column5, click **Insert new column** in Column 4.
12. To remove a row or column, click **Delete Row** or **Delete Column** for the specific row or column.
13. Right-click in the row/column you want to resize, point to **Grid**, and then select either **Column Width** or **Row Height**.

The Set Width or Set Height dialog box appears, depending on your selection.

14. To set the width/height:

Click...	To...
Automatic	Set the width/height to resize according to the largest object in the row/column.
Star	Set the width/height to a relational percentage of the space remaining in the grid after the size hierarchy is established.
Pixel	Set the width/height to a fixed number of pixels, regardless of the grid size or objects in the row/column.

- Or -

Hover the mouse cursor over a grid line to change the cursor to a double arrow, and then click and drag the grid line to the desired position.



Tip:

Click the **Toggle Width Type** button in the row/column you want to resize to switch between Star and Pixel sizing. You cannot size automatically using the **Toggle Width Type** button.



Note:

Resizing rows and columns in this manner is available only in design mode.

15. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Use a grid splitter

Grid splitter controls can only be used with grid controls.

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. Expand the control group you want to work with.
6. Click and drag the control to the desired position in the **Designer**.

**Note:**

To apply the properties of a control to another control, click and drag the control to another control in the document tree. The Applying **<control (control name)> to Target** dialog box appears where you can select the option you want to use to apply to the control.

7. Right-click the grid and select **Edit Layer**.
The grid's editor is initiated, which is indicated by an orange border. By default, there is one row (Row0) and one column (Column0) in the grid, and the row and column edit buttons are visible.
8. Click **Insert new row**.
A new row (Row1) appears with its own set of buttons.
9. Click **Insert new column**.
A new column (Column1) appears with its own set of buttons.
10. To add a row between existing rows, click **Insert new row** in the row above where you want to add the row. For example, if you want to add a new row between Row4 and Row5, click **Insert new row** in Row4.
11. To add a column between existing columns, click **Insert new column** in the column to the left of where you want to add the new column. For example, if you want to add a new column between Column4 and Column5, click **Insert new column** in Column 4.
12. To remove a row or column, click **Delete Row** or **Delete Column** for the specific row or column.
13. Right-click in the row/column you want to resize, point to **Grid**, and then select either **Column Width** or **Row Height**.
The Set Width or Set Height dialog box appears, depending on your selection.
14. To set the width/height:

Click...	To...
Automatic	Set the width/height to resize according to the largest object in the row/column.

Click...	To...
Star	Set the width/height to a relational percentage of the space remaining in the grid after the size hierarchy is established.
Pixel	Set the width/height to a fixed number of pixels, regardless of the grid size or objects in the row/column.

- Or -

Hover the mouse cursor over a grid line to change the cursor to a double arrow, and then click and drag the grid line to the desired position.



Tip:

Click the **Toggle Width Type** button in the row/column you want to resize to switch between Star and Pixel sizing. You cannot size automatically using the **Toggle Width Type** button.



Note:

Resizing rows and columns in this manner is available only in design mode.

15. From the **Toolbox**, add a grid splitter in the column you want to be able to resize.

16. Configure the properties, as required.



Note:

To view the full list of properties, select the **Advanced** check box.

17. Set the position of the grid splitter using the **Horizontal Alignment** and **Vertical Alignment** properties.

18. To specify whether to resize only the row or column, set the **ResizeDirection** property.

19. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Use grid anchors

When a control is added to a grid, the control gets an extra set of handles called anchors. These four anchors appear outside the control's blue resize handles on the top, bottom, and sides, and assist with positioning objects within the grid.

Before you begin



Note:

The blue dots are the resize handles; the black/grey dots are the anchors. Black anchors indicate that the control is anchored into position. By default, the top and left anchors are set.

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. Expand the control group you want to work with.
6. Click and drag the control to the desired position in the **Designer**.



Note:

To apply the properties of a control to another control, click and drag the control to another control in the document tree. The Applying **<control (control name)>** to **Target** dialog box appears where you can select the option you want to use to apply to the control.

7. Click the black anchors to unlock the control's position.
8. Move the control to the desired position, and then click the anchors to lock the position.
9. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Border



The Border control is a decorator typically used to draw a curved border around another container.

A border can accept one child object and serves as a base for other layout objects that do not have a border and border thickness properties, such as a canvas or a grid, or for styles and control templates.

See **Grids and Panels** to view the usage and description of the properties for this control.

Related reference

[Grids and Panels \(on page 846\)](#)

Canvas



The Canvas panel explicitly positions objects starting from the canvas' upper left-hand corner, according to exact X,Y coordinates.

Within a canvas, child controls are rendered sequentially by the order of their XAML markup. For example, if you draw two rectangles on a canvas and move the first rectangle on top of the second, the first rectangle appears underneath, or behind, the second rectangle. Since this rectangle was drawn, and therefore rendered, first, it cannot layer itself on top of the next objects drawn on the canvas.

The Canvas panel renders its child controls more efficiently than the dock or stack panel; however, it cannot resize itself.

See **Grids and Panels** to view the usage and description of the properties for this control.

Related reference

[Grids and Panels \(on page 846\)](#)

Data Grid



The Data Grid is a WPF control that uses a DataTable as input to populate a two-dimensional grid and to automatically determine its appearance. You can also hide columns on the input table that are not required.

A data grid's rows can be selected and individual cells can be edited. As a result, the two output DataTable properties are automatically populated: SelectedRows and OutputTable. The SelectedRows property outputs the selected row(s) as a new table. The OutputTable property displays the input table after it has been edited.

See **Grids and Panels** to view the usage and description of the properties for this control.

Related reference

[Grids and Panels \(on page 846\)](#)

Dock Panel



The Dock Panel is a WPF control that arranges its child controls horizontally or vertically, relative to each other, in a defined area.

A dock panel's children inherit an attached dependency property called Dock. This property appears in the object's Properties panel after the object has been added to a dock panel. The Dock property determines whether the objects are placed above, below, to the left, or to the right of one another inside the dock panel, depending on how the Dock property is set for each child control. By default, an object's Dock property is set to Left.



Note:

A dock panel does not explicitly position objects in the same way as the canvas or grid controls. Controls cannot be drawn onto the dock panel or selected and moved; they can only be positioned in relation to another object.

The Dock panel has a unique property called LastChildFill. This property specifies whether the last object added to the Dock panel fills the remaining space.

By default, the LastChildFill property is enabled. When enabled, the last control added to the Dock panel automatically appears stretched in the remaining space of the dock panel, regardless of how the Dock property is set on the control. For the last control's Dock property to work, the LastChildFill check box property on the Dock panel must be cleared.

If you want the control added to your dock panel to fill the remaining space or stretch within the dock panel, set the control's Width and Height properties to Auto.

See **Grids and Panels** to view the usage and description of the properties for this control.

Related reference

[Grids and Panels \(on page 846\)](#)

Grid



The Grid panel consists of adaptable rows and columns and has resizing capabilities for itself and its child controls. Using the Margin property, you can set precise spacing for columns and rows.

When a grid panel is set as the active layer, an orange border indicates that you can add additional objects.

When a grid is the active layer, headers appear across the top and sides of the grid to define its rows and columns. By default, the grid has one row and one column and is zero-based; that is, the first row is Row 0 and the first column is Column 0. The grid panel provides three editing buttons for adding rows and columns, deleting rows and columns, and changing the row/column width type.

Grid Anchors

When an object is added to a grid, the object receives a set of anchors. These four anchors appear just above the object's blue move/resize handles, on the object's top, bottom and sides. These grid anchors allow you to position objects within the grid by attaching the object to the top, bottom, or sides of the grid cell. Attaching the object to the grid cell keeps the object in that position when the grid is resized.

Grid anchors are either black or grey. A black anchor indicates that the object is locked in the current position, whereas a grey anchor indicates the object is unlocked and can be repositioned. Clicking the grid anchors changes them from black to grey and vice versa. By default, an object is locked to the top and left side of a grid cell.

Sizing Rows and Columns

The Forms Designer supports advanced editing of grid cell height and width sizing. This functionality is accessed through the grid's context menu. You can choose automatic, Star (relational), or pixel resizing.

See **Grids and Panels** to view the usage and description of the properties for this control.

Related reference

[Grids and Panels \(on page 846\)](#)

Grid Splitter



The Grid Splitter is used to automatically add resize behavior to a row or column in a grid.

See **Grids and Panels** to view the usage and description of the properties for this control.

Related reference

[Grids and Panels \(on page 846\)](#)

Ink Presenter



The Ink Presenter is a Silverlight control that displays ink on a surface.

See **Grids and Panels** to view the usage and description of the properties for this control.

Related reference

[Grids and Panels \(on page 846\)](#)

Stack Panel



The Stack Panel is a WPF control that is similar to a dock panel, except that it stacks its child controls in either a horizontal or vertical single line, depending on its orientation, in a defined area. Stack panels are not capable of wrapping their child controls. Controls are added to a stack panel in the same way as a dock panel.

By default, objects are stacked vertically. You can use the Orientation property to re-orient a stack panel.

Although the stack panel cannot anchor its child controls like the grid panel, you can emulate object anchoring for proper resizing. To do this, first set the Width and Height properties of the stack panel to Auto. As long as the stack panel is placed inside a root element with resizing capabilities—for example, a grid—the entire stack panel resizes with its root element. Then, depending on the orientation of the stack panel, change either the height (for horizontal orientation) or width (for vertical orientation) of the child control(s) to Auto. This anchors the objects to the edges of the stack panel and resizes the child objects when the root element is resized.

See **Grids and Panels** to view the usage and description of the properties for this control.

Related reference

[Grids and Panels \(on page 846\)](#)

Uniform Grid



The Uniform Grid is a WPF control that uses an invisible grid-type layout to evenly space and arrange its child controls. Controls are added to a uniform grid in the same manner as they are added to a dock panel.

By specifying a set number of rows and columns for the grid with the panel's Rows and Columns properties, controls are sequentially added and evenly spaced according to the grid's size and number of rows and columns. By default, the Forms Designer creates uniform grids with five rows and five columns. Additionally, the panel's FlowDirection property determines which side controls are added from; that is, from the left or the right side.

Unlike the regular grid panel, the uniform grid does not allow you to edit row and column height or width. However, this panel automatically readjusts its child controls on the workspace when it is resized; that is, you do not have to perform a test run to show how it resizes its contents. But, it does not alter a control's width or height as all controls are, by default, anchored to the left side of the panel.

See **Grids and Panels** to view the usage and description of the properties for this control.

Related reference

[Grids and Panels \(on page 846\)](#)

View Box



The View Box is a container that can scale its content to any size. It has a single child property. If you need the contents of a canvas to scale, it can be added as a child of a view box to get the desired effect.



Note:

The Forms Designer may not be able to properly position the objects if you nest view boxes.

See **Grids and Panels** to view the usage and description of the properties for this control.

Related reference[Grids and Panels \(on page 846\)](#)

Viewport 3D



The Viewport 3D panel is a WPF control that provides a rendering surface for three-dimensional visual content while providing properties consistent with two-dimensional layouts, such as clipping, height and width, and mouse events.

When the Viewport 3D control is included as the content of a layout element, such as Canvas, you must specify the size of the Viewport 3D by setting its Height and Width properties.

See **Grids and Panels** to view the usage and description of the properties for this control.

Related reference[Grids and Panels \(on page 846\)](#)

Wrap Panel



The Wrap Panel is a WPF control that arranges child controls sequentially and linearly from left to right within a defined area and wraps (that is, breaks) to the next line when a control reaches the edge of the defined area.

As with a dock or stack panel, wrap panels have either a horizontal or vertical orientation, which is adjusted with their Orientation property, and adds controls from either the left or right side, depending upon their FlowDirection property. Controls are added to a wrap panel in the same manner as they are added to a dock panel.

The wrap panel's ItemWidth and ItemHeight properties determine the child control's size and layout. By default, these properties are set to Auto.

Since this panel automatically readjusts its child controls in the designer when it is resized, it is easy to visualize how this panel rearranges its contents when placed within a root element with resizing capabilities. For example, consider a wrap panel with horizontal orientation that contains three side-by-side buttons. When the panel is resized to where the edge of the third button meets the right side of the panel, the third button is automatically "wrapped" or shifted downward to the next line to accommodate the panel's new size.

See **Grids and Panels** to view the usage and description of the properties for this control.

Related reference[Grids and Panels \(on page 846\)](#)

Indicator Controls

Indicator controls are used to graphically display states and values. A standard indicator has an on and off state and is not dependent on a numeric value. A level indicator displays the state associated with the value assigned to the control.

When an Indicator control receives a Null value for the Value property, the control sets its Opacity property to 0.2 until a valid Value is received. The control appears disabled until a valid Value is received.

Indicator Controls Property Usage

The following table shows the properties used by each control:

Property	Level Indica- tor Five State	Level Indica- tor Three State	Tower Indica- tor Five State	Tower Indica- tor Three State
AllowDrop	X	X	X	X
Background	X	X	X	X
BorderBrush	X	X	X	X
BorderThickness	X	X	X	X
ClipToBounds	X	X	X	X
Column	X	X	X	X
ColumnSpan	X	X	X	X
CornerRadius	X	X	X	X
Effect	X	X	X	X
FlashDelay	X	X	X	X
FlashState1	X	X	X	X
FlashState2	X	X	X	X
FlashState3	X	X	X	X
FlashState4	X		X	
FlashState5	X		X	

Property	Level Indica- tor Five State	Level Indica- tor Three State	Tower Indica- tor Five State	Tower Indica- tor Three State
FlowDirection	X	X	X	X
Focusable	X	X	X	X
FocusVisualStyle	X	X	X	X
FontFamily	X	X	X	X
FontSize	X	X	X	X
FontStretch	X	X	X	X
FontStyle	X	X	X	X
FontWeight	X	X	X	X
ForceCursor	X	X	X	X
Foreground	X	X	X	X
GlareBrush	X	X	X	X
Height	X	X	X	X
HorizontalAlign- ment	X	X	X	X
HorizontalContent- Alignment	X	X	X	X
InnerBorderBrush	X	X	X	X
InnerBorderThick- ness	X	X	X	X
InnerCornerRadius	X	X	X	X
IsEnabled	X	X	X	X
IsHitTestVisible	X	X	X	X
IsManipulationEn- abled	X	X	X	X
IsSharedSizeScope	X	X	X	X
IsTabStop	X	X	X	X
Localization	X	X	X	X

Property	Level Indica- tor Five State	Level Indica- tor Three State	Tower Indica- tor Five State	Tower Indica- tor Three State
Margin	X	X	X	X
MaxHeight	X	X	X	X
MaxWidth	X	X	X	X
MinHeight	X	X	X	X
MinorBackground	X	X	X	X
MinWidth	X	X	X	X
Name	X	X	X	X
Opacity	X	X	X	X
OpacityMask	X	X	X	X
OverridesDefault- Style	X	X	X	X
Padding	X	X	X	X
RenderTransform- Origin	X	X	X	X
Row	X	X	X	X
RowSpan	X	X	X	X
SnapsToDevicePix- els	X	X	X	X
State1Brush	X	X	X	X
State2Brush	X	X	X	X
State2IsOn			X	X
State2Threshold	X	X	X	X
State3Brush	X	X	X	X
State3IsOn			X	X
State3Threshold	X	X	X	X
State4Brush	X		X	

Property	Level Indica- tor Five State	Level Indica- tor Three State	Tower Indica- tor Five State	Tower Indica- tor Three State
State4IsOn			X	
State4Threshold	X		X	
State5Brush	X		X	
State5IsOn			X	
State5Threshold	X		X	
Style	X	X	X	X
TabIndex	X	X	X	X
Tag	X	X	X	X
Template	X	X	X	X
TertiaryBroder- Brush	X	X	X	X
TertiaryBorder- Thickness	X	X	X	X
TertiaryCornerRa- dius	X	X	X	
ToolTip	X	X	X	X
Uid	X	X	X	X
UseLayoutRound- ing	X	X	X	X
Value	X	X	X	X
VerticalAlignment	X	X	X	X
VerticalContentAl- ignment	X	X	X	X
Visibility	X	X	X	X
Width	X	X	X	X

Indicator Controls Property Descriptions

The following table shows the properties descriptions used by all control:

Property	Description
AllowDrop	Select this check box to indicate that the control can accept data that is dragged onto it.
Background	Expand this field to select the background color of the selected control.
BorderBrush	Expand this field to select the border color of the selected control.
BorderThickness	Specifies the border thickness of a control, in pixels.
ClipToBounds	Select this check box to clip the content of this control (or content coming from the child elements of this control) to fit into the size of the containing control.
Column	Specifies the column within the grid where the control is located.
ColumnSpan	Specifies the total number of columns that child content spans within a grid.
CornerRadius	Specifies the degree to which the corners of a control are rounded.
Effect	<p>Click the arrow to expand the property, select the effect you want to apply to the control, and then configure the specific values for the effect. Valid values to select are:</p> <ul style="list-style-type: none"> • Blur <ul style="list-style-type: none"> ◦ Kernel Type ◦ Radius • Drop Shadow <ul style="list-style-type: none"> ◦ Color ◦ Direction ◦ Opacity ◦ Shadow Depth

Property	Description
FlashDelay	Specifies the amount of time, in milliseconds, that the indicator light waits, while being pressed, before a command to move the Thumb executes.
FlashState1	Select this check box to indicate that State1 is active.
FlashState2	Select this check box to indicate that State2 is active.
FlashState3	Select this check box to indicate that State3 is active.
FlashState4	Select this check box to indicate that State4 is active.
FlashState5	Select this check box to indicate that State5 is active.
FlowDirection	<p>Specifies the direction that text and other user interface (UI) elements flow within any parent control. Selections are:</p> <ul style="list-style-type: none"> • Left to Right • Right to Left
Focusable	Select this check box to indicate that the control can receive focus.
FocusVisualStyle	Specifies a property that enables customization of appearance, effects, or other style characteristics that apply to this control when it has focus.
FontFamily	Specifies the font family for the control.
FontSize	Specifies the font size for the control.
FontStretch	Specifies the degree to which a font is condensed or expanded on the screen.
FontStyle	Specifies the font style for the control.
FontWeight	Specifies the weight or thickness of the specified font.

Property	Description
ForceCursor	Select this check box to indicate that the user interface (UI) renders the cursor as declared by the Cursor property.
Foreground	Expand this field to select the foreground color.
GlareBrush	Expand this field to specify the glare effect for the control.
Height	Specifies the height of the control, in pixels.
HorizontalAlignment	<p>Specifies the horizontal alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
HorizontalContentAlignment	<p>Specifies the horizontal alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
ImageStretch	<p>Specifies how the image should be stretched to fill the destination rectangle. Selections are:</p> <ul style="list-style-type: none"> • Fill • None • Uniform • UniformToFill
InnerBorderBrush	Expand this field to select the color for the inner border.
InnerBorderThickness	Expand this field to select the color for the inner border.

Property	Description
InnerCornerRadius	Specifies the degree to which the corners of the inner border are rounded.
IsEnabled	Select this check box to indicate that this control is enabled in the user interface (UI).
IsHitTestVisible	Select this check box to indicate that the control can possibly be returned as a hit test result from some portion of its rendered content.
IsManipulationEnabled	Select this check box to indicate that the manipulation events for the control are enabled.
IsSharedSizeScope	Select this check box to indicate that controls are sharing size information.
IsTabStop	Select this check box to indicate that a control is included in tab navigation.
Localization	Click the ellipsis button to open the Localization Configuration dialog box where you can configure the property to localize, the language to use for localization, and the text to localize.
Margin	Specifies the outer margin of a control.
MaxHeight	Specifies the maximum height of a control.
MaxWidth	Specifies the maximum width of a control.
MinHeight	Specifies the minimum height of a control.
MinorBackground	Expand this field to select the fill color of secondary components on the control.
MinWidth	Specifies the minimum width of a control.
Name	Specifies the identifying name of the control. The name provides a reference so that code behind, such as event handler code, can refer to a markup element during processing by a XAML processor.

Property	Description
Opacity	Specifies the opacity factor applied to the entire control. Valid values are 0 through 1, with up to 15 decimal places.
OpacityMask	Expand this field to select a color for the opacity mask, which is applied to any alpha-channel masking for the rendered content of this control.
OverridesDefaultStyle	Select this check box to indicate that this control incorporates style properties from theme styles.
Padding	Specifies the amount of padding to apply to the inside edge of each side of a control.
RenderTransformOrigin	<p>Specifies the center point of any possible render transform, relative to the bounds of the element.</p> <ul style="list-style-type: none"> • X is the horizontal origin factor, given as a value between 0 and 1. • Y is the vertical origin factor, given as a value between 0 and 1.
Row	RowBackground
RowSpan	Specifies the total number of rows that child content spans within a grid.
SnapsToDevicePixels	Select this check box to indicate that rendering for this control uses device-specific pixel settings.
State1Brush	Expand this field to select a color for the indicator light when it is in State1.
State2Brush	Expand this field to select a color for the indicator light when it is in State2.
State2IsOn	Select this check box to indicate that State2 is active.
State2Threshold	Specifies, in a numerical value, the maximum number that State1 will reach before changing to State2.

Property	Description
State3Brush	Expand this field to select a color for the indicator light when it is in State3.
State3IsOn	Select this check box to indicate that State3 is active.
State3Threshold	Specifies, in a numerical value, the maximum number that State2 will reach before changing to State3.
State4Brush	Expand this field to select a color for the indicator light when it is in State4.
State4IsOn	Select this check box to indicate that State4 is active.
State4Threshold	Specifies, in a numerical value, the maximum number that State3 will reach before changing to State4.
State5Brush	Expand this field to select a color for the indicator light when it is in State5.
State5IsOn	Select this check box to indicate that State5 is active.
State5Threshold	Specifies, in a numerical value, the maximum number that State4 will reach before changing to State5.
Style	Specifies the style used by this control when it is rendered.
TabIndex	Specifies the order in which controls receive focus when the Tab key is pressed.
Tag	Specifies an arbitrary object value that can be used to store custom information about this control.
Template	Click the ellipsis button to open the Open dialog box to select a template to use.

Property	Description
TertiaryBorderBrush	Expand this field to select the innermost border color of the selected control.
TertiaryBorderThickness	Specifies the thickness of the innermost border; that is, the border closest to the indicator.
TertiaryCornerRadius	Specifies the degree to which the innermost corners of a control are rounded.
ToolTip	Specifies the tool-tip that displays for this control in the user interface (UI).
Uid	Specifies the unique identifier (for localization) for this control.
UseLayoutRounding	Specifies a value that determines whether rendering for the object and its visual subtree should use rounding behavior that aligns rendering to whole pixels.
Value	Specifies the current magnitude of the range control.
VerticalAlignment	<p>Specifies the vertical alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
VerticalContentAlignment	<p>Specifies the vertical alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
Visibility	Specifies the visibility of this control. Selections are:

Property	Description
	<ul style="list-style-type: none"> • Collapsed • Hidden • Visible
Width	Specifies the width of the control, in pixels.

Level Indicator Five State



The Level Indicator Five State control is a panel indicator with five possible state colors and four possible threshold values. The state is determined by the value assigned to the control.

See **Indicator Controls** to view the usage and description of the properties for this control.

Related reference

[Indicator Controls \(on page 881\)](#)

Level Indicator Three State



The Level Indicator Three State control is a panel indicator with three possible state colors and two possible threshold values. The state is determined by the value assigned to the control.

See **Indicator Controls** to view the usage and description of the properties for this control.

Related reference

[Indicator Controls \(on page 881\)](#)

Tower Indicator Five State



The Tower Indicator Five State control is a set of five indicators assembled in a tower. There are five possible state colors and four possible threshold values. The five indicators function independently and can monitor five separate processes.

See **Indicator Controls** to view the usage and description of the properties for this control.

Related reference

[Indicator Controls \(on page 881\)](#)

Tower Indicator Three State



The Tower Indicator Three State control is a set of three indicators assembled in a tower. There are three possible state colors and two possible threshold values. The three indicators function independently and can monitor three separate processes.

See **Indicator Controls** to view the usage and description of the properties for this control.

Related reference

[Indicator Controls \(on page 881\)](#)

Input Controls

Input controls are used to input data to write to a server.

Input Controls Property Usage

The following table shows the properties used by each control:

Property	Data Text Box
AcceptsReturn	X
AcceptsTab	X
AllowDrop	X
AutoWordSelection	X
Background	X
BorderBrush	X
BorderThickness	X
CaretBrush	X
CharacterCasing	X
ClipToBounds	X
Column	X
ColumnSpan	X
Effect	X

Property	Data Text Box
FlowDirection	X
Focusable	X
FocusVisualStyle	X
FontFamily	X
FontSize	X
FontStretch	X
FontStyle	X
FontWeight	X
ForceCursor	X
Foreground	X
Height	X
HorizontalAlignment	X
HorizontalContentAlignment	X
HorizontalScrollBarVisibility	X
IsEnabled	X
IsHitTestVisible	X
IsManipulationEnabled	X
IsReadOnly	X
IsReadOnlyCaretVisible	X
IsSharedSizeScope	X
IsTabStop	X
IsUndoEnabled	X
Localization	X
Margin	X
MaxHeight	X

Property	Data Text Box
MaxLength	X
MaxLines	X
MaxWidth	X
MinHeight	X
MinLines	X
MinWidth	X
Name	X
Opacity	X
OpacityMask	X
OverridesDefaultStyle	X
Padding	X
RenderTransformOrigin	X
Row	X
RowSpan	X
SelectionBrush	X
SelectionOpacity	X
SnapsToDevicePixels	X
Style	X
TabIndex	X
Tag	X
Template	X
Text	X
TextAlignment	X
TextWrapping	X
ToolTip	X
Uid	X

Property		Data Text Box
UseLayoutRounding	X	
VerticalAlignment	X	
VerticalContentAlignment	X	
VerticalScrollBarVisibility	X	
Visibility	X	
Width	X	

Input Controls Property Descriptions

The following table shows the properties descriptions used by all control:

Property	Description
AcceptsReturn	Select this check box to indicate that pressing Enter in a multiline Text Box control creates a new line of text in the control or activates the default button.
AcceptsTab	Select this check box to indicate that pressing the Tab key in a multiline text box control types a tab character in the control instead of moving the focus to the next control in the tab order.
AllowDrop	Select this check box to indicate that the control can accept data that is dragged onto it.
AutoWordSelection	Select this check box to indicate that when you select part of a word by dragging across it with the mouse, the rest of the word is selected.
Background	Expand this field to select the background color of the selected control.
BorderBrush	Expand this field to select the border color of the selected control.
BorderThickness	Specifies the border thickness of a control, in pixels.

Property	Description
CaretBrush	Expand this field to select the brush to use to render the vertical bar that indicates the insertion point.
CharacterCasing	<p>Specifies how characters are cased when they are manually entered into the text box. Selections are:</p> <ul style="list-style-type: none"> • Normal • Lower • Upper
ClipToBounds	Select this check box to clip the content of this control (or content coming from the child elements of this control) to fit into the size of the containing control.
Column	Specifies the column within the grid where the control is located.
ColumnSpan	Specifies the total number of columns that child content spans within a grid.
Effect	<p>Click the arrow to expand the property, select the effect you want to apply to the control, and then configure the specific values for the effect. Valid values to select are:</p> <ul style="list-style-type: none"> • Blur <ul style="list-style-type: none"> ◦ Kernel Type ◦ Radius • Drop Shadow <ul style="list-style-type: none"> ◦ Color ◦ Direction ◦ Opacity ◦ Shadow Depth
FlowDirection	Specifies the direction that text and other user interface (UI) elements flow within any parent control. Selections are:

Property	Description
	<ul style="list-style-type: none"> • Left to Right • Right to Left
Focusable	Select this check box to indicate that the control can receive focus.
FocusVisualStyle	Specifies a property that enables customization of appearance, effects, or other style characteristics that apply to this control when it has focus.
FontFamily	Specifies the font family for the control.
FontSize	Specifies the font size for the control.
FontStretch	Specifies the degree to which a font is condensed or expanded on the screen.
FontStyle	Specifies the font style for the control.
FontWeight	Specifies the weight or thickness of the specified font.
ForceCursor	Select this check box to indicate that the user interface (UI) renders the cursor as declared by the Cursor property.
Foreground	Expand this field to select the foreground color.
Height	Specifies the height of the control, in pixels.
HorizontalAlignment	<p>Specifies the horizontal alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
HorizontalContentAlignment	<p>Specifies the horizontal alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Left • Center

Property	Description
	<ul style="list-style-type: none"> • Right • Stretch
HorizontalScrollBarVisibility	<p>Specifies whether a horizontal scroll bar is shown. Selections are:</p> <ul style="list-style-type: none"> • Auto - the scroll bar appears only if the content exceeds the width of the control • Disabled - the scroll bar never appears • Hidden - the scroll bar is hidden from view • Visible - the scroll bar is visible at all times
ImageStretch	<p>Specifies how the image should be stretched to fill the destination rectangle. Selections are:</p> <ul style="list-style-type: none"> • Fill • None • Uniform • UniformToFill
IsEnabled	<p>Select this check box to indicate that this control is enabled in the user interface (UI).</p>
IsHitTestVisible	<p>Select this check box to indicate that the control can possibly be returned as a hit test result from some portion of its rendered content.</p>
IsManipulationEnabled	<p>Select this check box to indicate that the manipulation events for the control are enabled.</p>
IsReadOnly	<p>Select this check box to indicate that the text editing control is read-only to a user interacting with the control.</p>
IsReadOnlyCaretVisible	<p>Specifies whether a read-only text box displays a caret. If the IsReadOnly check box is selected, you can still select and copy text. If the IsReadOnlyCaretVisible check box is also selected, a caret appears in the text box when the text box has keyboard focus. When the IsReadOnly check box</p>

Property	Description
	is cleared, selecting the <code>IsReadOnlyCaretVisible</code> check box has no effect.
<code>IsSharedSizeScope</code>	Select this check box to indicate that controls are sharing size information.
<code>IsTabStop</code>	Select this check box to indicate that a control is included in tab navigation.
<code>IsUndoEnabled</code>	Select this check box to indicate that undo support is enabled for the text-editing control. Clearing this check box clears the undo stack. Therefore, if you disable undo and then re-enable it, undo commands still do not work because the undo stack was emptied when you cleared the check box.
Localization	Click the ellipsis button to open the Localization Configuration dialog box where you can configure the property to localize, the language to use for localization, and the text to localize.
<code>Margin</code>	Specifies the outer margin of a control.
<code>MaxHeight</code>	Specifies the maximum height of a control.
<code>MaxLength</code>	Specifies the maximum length of text that can be entered in the control. When this property is set to 0, the maximum length of the text that can be entered in the control is limited only by available memory.
<code>MaxLines</code>	Specifies the maximum number of visible lines.
<code>MaxWidth</code>	Specifies the maximum width of a control.
<code>MinHeight</code>	Specifies the minimum height of a control.
<code>MinLines</code>	Specifies the minimum number of visible lines.
<code>MinWidth</code>	Specifies the minimum width of a control.
<code>Name</code>	Specifies the identifying name of the control. The name provides a reference so that code behind,

Property	Description
	such as event handler code, can refer to a markup element during processing by a XAML processor.
Opacity	Specifies the opacity factor applied to the entire control. Valid values are 0 through 1, with up to 15 decimal places.
OpacityMask	Expand this field to select a color for the opacity mask, which is applied to any alpha-channel masking for the rendered content of this control.
OverridesDefaultStyle	Select this check box to indicate that this control incorporates style properties from theme styles.
Padding	Specifies the amount of padding to apply to the inside edge of each side of a control.
RenderTransformOrigin	<p>Specifies the center point of any possible render transform, relative to the bounds of the element.</p> <ul style="list-style-type: none"> • X is the horizontal origin factor, given as a value between 0 and 1. • Y is the vertical origin factor, given as a value between 0 and 1.
Row	RowBackground
RowSpan	Specifies the total number of rows that child content spans within a grid.
SelectionBrush	Expand this field to select the brush to use to highlight the selected text.
SnapsToDevicePixels	Select this check box to indicate that rendering for this control uses device-specific pixel settings.
Style	Specifies the style used by this control when it is rendered.
TabIndex	Specifies the order in which controls receive focus when the Tab key is pressed.

Property	Description
Tag	Specifies an arbitrary object value that can be used to store custom information about this control.
Template	Click the ellipsis button to open the Open dialog box to select a template to use.
Text	Specifies the text contents of a control.
TextAlignment	<p>Specifies the horizontal alignment of text content. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Justify
TextWrapping	<p>Specifies how the control wraps text. Selections are:</p> <ul style="list-style-type: none"> • NoWrap • Wrap • WrapWithOverflow
ToolTip	Specifies the tool-tip that displays for this control in the user interface (UI).
Uid	Specifies the unique identifier (for localization) for this control.
UseLayoutRounding	Specifies a value that determines whether rendering for the object and its visual subtree should use rounding behavior that aligns rendering to whole pixels.
VerticalAlignment	<p>Specifies the vertical alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch

Property	Description
VerticalContentAlignment	<p>Specifies the vertical alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
VerticalScrollBarVisibility	<p>Specifies whether a vertical scroll bar is shown. Scroll bar selections are:</p> <ul style="list-style-type: none"> • Auto - appears only if the content exceeds the width of the control. • Disabled - never appears. • Hidden - hidden from view. • Visible - visible at all times.
Visibility	<p>Specifies the visibility of this control. Selections are:</p> <ul style="list-style-type: none"> • Collapsed • Hidden • Visible
Width	Specifies the width of the control, in pixels.

Data Text Box



The Data Text Box is a WPF control that is a simple two-way data binding control used for writing data to a server.

See **Input Controls** to view the usage and description of the properties for this control.

Related reference

[Input Controls \(on page 893\)](#)

Property	ComboBox	ComboBox [add]	Group Box	Items Control	List Box	List Box [add]	List View	List View [add]	MenuItem	MenuItem [add]	Separator	Status Bar	Status Bar [add]	Tab	Tab Item [add]	ToolBar	ToolBar Tray	TreeView	TreeView [add]
Is-Overflow-Open																X			
Is-ReadOnly	X																		
Is-Selected		X				X		X							X				X
Is-Shared-Size-Scope	X	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Is-Submenu-Open									X										
Is-Synchronized-With-Current-Item	X				X		X							X					

Property	ComboBox	ComboBox Item [add]	Group Box	Items Control	List Box	List Box Item [add]	List View Item [add]	List View Item Menu Item [add]	Menu Item [add]	Separator	Status Bar Item [add]	Status Bar Item [add]	Tab	Tab Item [add]	Tool Bar	Tool Bar Tray	Tree View Item [add]	Tree View Item [add]
Is-Tab-Stop	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X
Is-Text-Search-Case-Sensitive	X				X			X	X		X		X		X		X	X
Is-Text-Search-Enabled	X				X			X	X		X		X		X		X	X
Item-Panel				X														
Items	X																	
Item-Source				X	X		X	X			X		X		X		X	X
Item-String-Format					X		X	X			X		X		X		X	X

Prop-er-ty	Com-bo Box	Com-bo Group Item Box [add]	Items Con-trol	List Box	List Box Item [add]	List View Item [add]	List View Item [add]	Menu Item [add]	Menu Separator	Sta-tus Bar	Sta-tus Bar Item [add]	Tab	Tab Item [add]	Tool Bar	Tool Bar Tray	Tree View	Tree View Item [add]
Over-rides-Default-Style	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X
Padding	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X
Pro-jection			X														
Ren-der-Trans-form-Ori-gin	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Row	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Row-S-pan	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Select-ed-Index				X		X						X					
Select-ed-Item				X		X						X					
Select-ed-				X		X						X					

Property	ComboBox	ComboBox [add]	Group Box	Items Control	List Box	List Box [add]	List View	List View [add]	MenuItem	MenuItem [add]	Menu Separator	Status Bar	Status Bar [add]	Tab	Tab Item [add]	Tool Bar	Tool Bar Tray	TreeView	TreeView [add]
Tab-Strip-Placement														X					
Tag	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Template	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X
Text	X																		
ToolTip	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Uid	X	X	X		X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Use-Layout-Rounding	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Vertical-Alignment	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Vertical-Content-Alignment	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X		X	X

Property	ComboBox	ComboBox [add]	GroupBox	ItemsControl	ListBox	ListBox [add]	ListView	ListView [add]	MenuItem	MenuItem [add]	Separator	StatusBar	StatusBar [add]	TabItem	TabItem [add]	ToolBar	ToolBar Tray	TreeView	TreeView [add]
Visibility	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Width	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Items Controls Property Descriptions

The following table shows the properties descriptions used by all control:

Property	Description
AllowDrop	Select this check box to indicate that the control can accept data that is dragged onto it.
AlternationCount	Specifies the number of alternating item containers or rows to display.
Background	Expand this field to select the background color of the selected control.
Band	Specifies where the toolbar should be located in the toolbar tray.
BandIndex	Specifies the band index number that indicates the position of the toolbar on the band.
BorderBrush	Expand this field to select the border color of the selected control.
BorderThickness	Specifies the border thickness of a control, in pixels.
CacheMode	Specifies a value that indicates that rendered content should be cached when possible.
CharacterSpacing	Specifies the distance between characters of text in the control, measured in 1000ths of the font size.

Property	Description
Clip	<p>Specifies the geometric shape used to define the outline of the image. Selections are:</p> <ul style="list-style-type: none"> • Hover • Press • Release
ClipToBounds	<p>Select this check box to clip the content of this control (or content coming from the child elements of this control) to fit into the size of the containing control.</p>
Column	<p>Specifies the column within the grid where the control is located.</p>
ColumnSpan	<p>Specifies the total number of columns that child content spans within a grid.</p>
CommandParameter	<p>Specifies the parameter to pass to the Command property.</p>
CommandTarget	<p>Specifies the element on which to raise the specified command.</p>
Content	<p>Specifies the content displayed on the control.</p>
ContentStringFormat	<p>Specifies a composite string that determines how to format the Content property if it is displayed as a string.</p>
Cursor	<p>Specifies the cursor image to display when the mouse pointer hovers over an object.</p>
DataContext	<p>Specifies the data context for a control when it participates in data binding.</p>
DisplayMemberPath	<p>Specifies a path to a value on the source control to serve as the visual representation of the control.</p>
Effect	<p>Click the arrow to expand the property, select the effect you want to apply to the control, and then configure the specific values for the effect. Valid values to select are:</p>

Property	Description
	<ul style="list-style-type: none"> • Blur <ul style="list-style-type: none"> ◦ Kernel Type ◦ Radius • Drop Shadow <ul style="list-style-type: none"> ◦ Color ◦ Direction ◦ Opactiy ◦ Shadow Depth
FlowDirection	<p>Specifies the direction that text and other user interface (UI) elements flow within any parent control. Selections are:</p> <ul style="list-style-type: none"> • Left to Right • Right to Left
Focusable	<p>Select this check box to indicate that the control can receive focus.</p>
FocusVisualStyle	<p>Specifies a property that enables customization of appearance, effects, or other style characteristics that apply to this control when it has focus.</p>
FontFamily	<p>Specifies the font family for the control.</p>
FontSize	<p>Specifies the font size for the control.</p>
FontStretch	<p>Specifies the degree to which a font is condensed or expanded on the screen.</p>
FontStyle	<p>Specifies the font style for the control.</p>
FontWeight	<p>Specifies the weight or thickness of the specified font.</p>
ForceCursor	<p>Select this check box to indicate that the user interface (UI) renders the cursor as declared by the Cursor property.</p>
Foreground	<p>Expand this field to select the foreground color.</p>
Header	<p>Specifies the header text to display on the control.</p>

Property	Description
HeaderStringFormat	Specifies a composite string that indicates how to format the header if it is displayed as a string
Height	Specifies the height of the control, in pixels.
HorizontalAlignment	<p>Specifies the horizontal alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
HorizontalContentAlignment	<p>Specifies the horizontal alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
Icon	Specifies the icon that displays for a Menu Item.
ImageStretch	<p>Specifies how the image should be stretched to fill the destination rectangle. Selections are:</p> <ul style="list-style-type: none"> • Fill • None • Uniform • UniformToFill
IsCheckable	Select this check box to indicate that a Menu Item can be checked by default.
IsChecked	Select this check box to indicate that the control is checked.
IsDropDownOpen	Select this check box to indicate that the drop-down area for a combo box is open.
IsEditable	Select this check box to indicate that the text in the Combo Box can be edited.

Property	Description
IsEnabled	Select this check box to indicate that this control is enabled in the user interface (UI).
IsExpanded	Select this check box to indicate that the expander content window is visible.
IsHitTestVisible	Select this check box to indicate that the control can possibly be returned as a hit test result from some portion of its rendered content.
IsLocked	Select this check box to indicate that a Tool Bar control cannot be moved inside a Tool Bar Tray.
IsMainMenu	Specifies whether the menu control gains focus if the F10 or Alt key is pressed on the keyboard. If there are multiple menu controls in a form, then only one of the controls should have its check box selected; the IsMainMenu property check box for the other menu control(s) should be cleared.
IsManipulationEnabled	Select this check box to indicate that the manipulation events for the control are enabled.
IsOverflowOpen	Select this check box to indicate that the Tool Bar overflow area is visible.
IsReadOnlyCaretVisible	Specifies whether a read-only text box displays a caret. If the IsReadOnly check box is selected, you can still select and copy text. If the IsReadOnlyCaretVisible check box is also selected, a caret appears in the text box when the text box has keyboard focus. When the IsReadOnly check box is cleared, selecting the IsReadOnlyCaretVisible check box has no effect.
IsSelected	Select this check box to indicate that the item is selected.
IsSharedSizeScope	Select this check box to indicate that controls are sharing size information.

Property	Description
IsSubmenuOpen	Select this check box to indicate that the submenu of the Menu Item is open.
IsSynchronizedWithCurrentItem	Select this check box to indicate that a selector should keep the selected item synchronized with the current item in the Items property.
IsTabStop	Select this check box to indicate that a control is included in tab navigation.
IsTextSearchCaseSensitive	Select this check box to indicate that text searches are case sensitive.
IsTextSearchEnabled	Select this check box to indicate that text search is enabled on the control.
ItemPanel	Specifies the template that defines the panel that controls the layout of items.
Items	Specifies the collection used to generate the content of the control.
ItemsSource	Specifies the collection used to generate the content of the control.
ItemStringFormat	Specifies a composite string that determines how to format the items in the control if they are displayed as strings.
ItemTemplate	Click the ellipsis button to open the Open dialog box to select the data template used to display each item.
Language	Specifies the localization language that applies to an object.
Localization	Click the ellipsis button to open the Localization Configuration dialog box where you can configure the property to localize, the language to use for localization, and the text to localize.
Margin	Specifies the outer margin of a control.

Property	Description
MaxDropDownHeight	Specifies the maximum height for a combo box drop-down.
MaxHeight	Specifies the maximum height of a control.
MaxWidth	Specifies the maximum width of a control.
MinHeight	Specifies the minimum height of a control.
MinWidth	Specifies the minimum width of a control.
Name	Specifies the identifying name of the control. The name provides a reference so that code behind, such as event handler code, can refer to a markup element during processing by a XAML processor.
Opacity	Specifies the opacity factor applied to the entire control. Valid values are 0 through 1, with up to 15 decimal places.
OpacityMask	Expand this field to select a color for the opacity mask, which is applied to any alpha-channel masking for the rendered content of this control.
OverridesDefaultStyle	Select this check box to indicate that this control incorporates style properties from theme styles.
Padding	Specifies the amount of padding to apply to the inside edge of each side of a control.
Projection	Specifies the 3-D effect to apply to the current object.
RenderTransformOrigin	<p>Specifies the center point of any possible render transform, relative to the bounds of the element.</p> <ul style="list-style-type: none"> • X is the horizontal origin factor, given as a value between 0 and 1. • Y is the vertical origin factor, given as a value between 0 and 1.
Row	RowBackground

Property	Description
RowSpan	Specifies the total number of rows that child content spans within a grid.
SelectedIndex	Specifies the index of the first item in the current selection, or returns negative one (-1) if the selection is empty.
SelectedItem	Specifies the first item in the current selection or returns null if the selection is empty.
SelectedValue	Specifies the value of the SelectedItem property, obtained by using the SelectedValuePath property.
SelectedValuePath	Specifies the path that is used to get the SelectedValue property from the SelectedItem property.
SnapsToDevicePixels	Select this check box to indicate that rendering for this control uses device-specific pixel settings.
StaysOpenOnEdit	Select this check box to indicate that a Combo Box that is open and displaying a drop-down control, remains open when you click in the text box.
Style	Specifies the style used by this control when it is rendered.
TabIndex	Specifies the order in which controls receive focus when the Tab key is pressed.
TabNavigation	Specifies a value that modifies how tabbing and the TabIndex property work for this control.
TabStripPlacement	<p>Specifies how tab headers align relative to the tab content. Selections are:</p> <ul style="list-style-type: none"> • Top • Right • Bottom
Tag	Specifies an arbitrary object value that can be used to store custom information about this control.

Property	Description
Template	Click the ellipsis button to open the Open dialog box to select a template to use.
Text	Specifies the text contents of a control.
ToolTip	Specifies the tool-tip that displays for this control in the user interface (UI).
Uid	Specifies the unique identifier (for localization) for this control.
UseLayoutRounding	Specifies a value that determines whether rendering for the object and its visual subtree should use rounding behavior that aligns rendering to whole pixels.
VerticalAlignment	<p>Specifies the vertical alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
VerticalContentAlignment	<p>Specifies the vertical alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
Visibility	<p>Specifies the visibility of this control. Selections are:</p> <ul style="list-style-type: none"> • Collapsed • Hidden • Visible
Width	Specifies the width of the control, in pixels.

Add a menu item to a combo or list box

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer > Toolbox tab**.
4. Choose one of the following methods to add an item to a combo or list box:
 - Right-click the control in the designer, and then select **Add ListBox Item** or **Add Combo Box Item**, depending on the control you added.
 - Select the control in the designer, and then from the Items Control list, click **Combo Item [add]** or **List Box Item [add]**, depending on the control you added.
 - Select the control in the designer, and then in the Properties panel, bind the **ItemsSource** property to the list of items to show.



Tip:

The DisplayMemberPath property can be used to show a specific property if a list of objects is bound to it. For example, if you are binding to a list of DirectoryResource objects, you can set the DisplayMemberPath property to DisplayName to show only the user friendly name of each DirectoryResource object.

- Open **Project > MainPage.xaml.cs**, and then edit the code as follows:

```
List<String>items=new List<String>()
items.Add("one");
items.Add("two");
items.Add("three");
ListBox1.ItemsSource = items;
```



Note:

Do not add an item using both the toolbox and the ItemSource in code behind. The form will not necessarily produce errors during compilation, but it will not work as desired at runtime.

5. Configure the properties, as required.



Note:

To view the full list of properties, select the **Advanced** check box.

6. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Sort items in a combo or list box or a list view

Although there is no property available that enables alphabetical sorting of contents, code can be used to sort them.

Before you begin

This is one method for programmatically sorting code alphabetically:

```
using System;

using System.Collections;

using System.Collections.Generic;

using System.Data;

using System.Windows;

using System.Windows.Controls;

using Proficy.Platform.Core.DisplayFramework;

using Proficy.Platform.Core.DisplayFramework.Forms;

using System.Windows.Data;

using System.ComponentModel;

namespace SortedSearchExample
{
    public partial class MainForm : FormBase
    {
        public MainForm()
        {
            InitializeComponent();

            SearchMethod.ExecutedSuccessfully += new System.EventHandler(this.SearchMethod_ExecutedSuccessfully);
        }

        private void SearchMethod_ExecutedSuccessfully(object sender, System.EventArgs e) {

            Sort("DisplayName", ListSortDirection.Ascending);

        }
    }
}
```

```
private void Sort(string sortBy, ListSortDirection direction)
{
    ICollectionView dataView = CollectionViewSource.GetDefaultView(ComboBox1.ItemsSource);

    if (dataView != null)
    {
        dataView.SortDescriptions.Clear();

        SortDescription sd = new SortDescription(sortBy, direction);

        dataView.SortDescriptions.Add(sd);

        dataView.Refresh();
    }
}
}
```

Add a menu item to a menu control

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. Expand the control group you want to work with.
6. Select the control in the Designer, and then from the **Items Control** list, click **Menu Item [add]**.



Tip:

Resize the menu control to view all of the menu items.

A menu item appears in the menu control each time you click the control in the **Toolbox**.

7. Configure the properties, as required.



Note:

To view the full list of properties, select the **Advanced** check box.

8. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Add a menu item to a menu item

The Designer allows you to nest menu items within another menu item. When menu items are nested, a right arrow automatically displays when the top-level menu item is clicked in run mode.

Before you begin

You must add a menu and at least one top-level menu item before you can nest menu items.

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. Expand the control group you want to work with.
6. Click and drag the control to the desired position in the **Designer**.



Note:

To apply the properties of a control to another control, click and drag the control to another control in the document tree. The Applying **<control (control name)>** to **Target** dialog box appears where you can select the option you want to use to apply to the control.

7. Select the control in the Designer, and then from the **Items Control** list, click **Menu Item [add]**.



Tip:

Resize the menu control to view all of the menu items.

A menu item appears in the menu control each time you click the control in the **Toolbox**.

8. In the **Document Tree**, right-click the menu item you want to add a nested menu item to, and then select **Add Menu Item**.

The new menu item is automatically nested under the selected menu item.

9. Configure the properties, as required.

**Note:**

To view the full list of properties, select the **Advanced** check box.

10. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Add a separator to an items control

Separators are commonly used with items controls, such as menus and menu items, toolbars, and status bars.

Before you begin

**Note:**

You must have added an item control with child items before adding a separator.

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. Expand the control group you want to work with.
6. Click and drag the control to the desired position in the **Designer**.

**Note:**

To apply the properties of a control to another control, click and drag the control to another control in the document tree. The Applying **<control (control name)> to Target** dialog box appears where you can select the option you want to use to apply to the control.

7. Add control items to the control, as required.
8. Select the control in the designer, and then from the **Items Control** list, click **Separator**.
9. Drag the separator control and drop it on the item control.
10. Click and drag the resize handles to size and position the separator.

**Tip:**

You can use the Ctrl + arrow buttons on your keyboard to position the separator more precisely.

11. Configure the properties, as required.

**Note:**

To view the full list of properties, select the **Advanced** check box.

12. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Add a status bar item to a status bar

This procedure applies only to status bar item controls.

Before you begin

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. Expand the control group you want to work with.
6. Click and drag the control to the desired position in the **Designer**.

**Note:**

To apply the properties of a control to another control, click and drag the control to another control in the document tree. The Applying **<control (control name)> to Target** dialog box appears where you can select the option you want to use to apply to the control.

7. Right-click the status bar control and click **Add StatusBar Item**.
A status bar item appears in the status bar control.

8. Configure the properties, as required.



Note:

To view the full list of properties, select the **Advanced** check box.

9. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Add a control to a status bar

This procedure applies to all controls other than a status bar item control.

Before you begin

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. Expand the control group you want to work with.
6. Click and drag the control to the desired position in the **Designer**.



Note:

To apply the properties of a control to another control, click and drag the control to another control in the document tree. The Applying **<control (control name)>** to **Target** dialog box appears where you can select the option you want to use to apply to the control.

7. Click the **Document Tree** tab.
8. From the **Toolbox**, click and drag a control to the document tree and drop it on the status bar control.
The Applying **<control (control name)>** to **Target** dialog box appears.
9. Select the **As a Child** option, and then click **OK**.
10. Configure the properties, as required.

**Note:**

To view the full list of properties, select the **Advanced** check box.

11. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Add a tab item to a tab control

Before you begin

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. Expand the control group you want to work with.
6. Click and drag the control to the desired position in the **Designer**.

**Note:**

To apply the properties of a control to another control, click and drag the control to another control in the document tree. The Applying **<control (control name)>** to **Target** dialog box appears where you can select the option you want to use to apply to the control.

7. Choose one of the following ways to add a tab item to the tab control:
 - Right-click the tab control in the Document Tree and click **Add Tab Item**. A tab item appears in the tab control.
 - Select the tab control in the canvas, right-click, and select **Add Tab Item**.
 - Select the tab control in the canvas, and in the Toolbox click **Items Control > Tab Item [Add]**

**Note:**

To apply the properties of a control to another control, click and drag the control to another control in the document tree. The Applying <control (control name)> to Target dialog box appears where you can select the option you want to use to apply to the control.

8. Configure the properties, as required.

**Note:**

To view the full list of properties, select the **Advanced** check box.

9. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Add a control to a tab item

This procedure provides an example of how to add a control to a tab item.

Before you begin

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. Add a tab control with at least one tab item.
6. Add a grid (or other container control) to the top-level of the designer, and then add the desired controls to it. Set the tab item as the active layer, and then move the grid control to the tab item within the document tree or by copying and using the Paste Special feature.
7. Configure the properties, as required.

**Note:**

To view the full list of properties, select the **Advanced** check box.

- Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Related reference

[Shapes \(on page 1057\)](#)

Related information

[Designer: Control Toolbox \(on page 725\)](#)

Add a control to a toolbar

Before you begin

Procedure

- In the navigator, click **Global Displays > Forms**.
- From the **Forms** list, select the resource you want to work with.
- Click **Display Editor > Designer**.
- Click the **Toolbox** tab.
- Expand the control group you want to work with.
- Click and drag the control to the desired position in the **Designer**.

**Note:**

To apply the properties of a control to another control, click and drag the control to another control in the document tree. The Applying **<control (control name)>** to **Target** dialog box appears where you can select the option you want to use to apply to the control.

- Click the **Document Tree** tab.
- From the **Toolbox**, click and drag a control to the document tree and drop it on the toolbar control. The Applying **<control (control name)>** to Target dialog box appears.

9. Select the **As a Child** option, and then click **OK**.
10. Configure the properties, as required.

**Note:**

To view the full list of properties, select the **Advanced** check box.

11. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Add a toolbar to a toolbar tray

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. Expand the control group you want to work with.
6. Click and drag the control to the desired position in the **Designer**.

**Note:**

To apply the properties of a control to another control, click and drag the control to another control in the document tree. The Applying **<control (control name)>** to **Target** dialog box appears where you can select the option you want to use to apply to the control.

7. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

8. From the **Toolbox**, click and drag a control to the document tree and drop it on the toolbar control.

The Applying <control (control name)> to Target dialog box appears.

9. Select the **As a Child** option, and then click **OK**.
10. Configure the properties, as required.



Note:

To view the full list of properties, select the **Advanced** check box.

11. Add a Tool Bar Tray control to the form.
12. Right-click the Tool Bar control, and then select **Cut**.
13. In the **Document Tree**, right-click the **Toolbar Tray** control and select **Paste Special**.
The Applying <control (control name)> to Target dialog box appears.
14. Select the **As a Child** option, and then click **OK**.



Tip:

If there are space constraints on the form, you can resize the toolbar tray to hide entire toolbar controls or the controls on the toolbar control. When you use the toolbar tray in run time, the hidden controls can be expanded using the drop-down arrow on the toolbar.

15. Configure the properties, as required.



Note:

To view the full list of properties, select the **Advanced** check box.

16. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Add a tree view item to a tree view

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.

4. Click the **Toolbox** tab.
5. Expand the control group you want to work with.
6. Click and drag the control to the desired position in the **Designer**.

**Note:**

To apply the properties of a control to another control, click and drag the control to another control in the document tree. The Applying **<control (control name)>** to **Target** dialog box appears where you can select the option you want to use to apply to the control.

7. Select the control in the designer, and then from the **Items Control** list, click **TreeView Item [add]**.

**Tip:**

Resize the tree view control to view all of the tree view items.

A tree view item appears in the tree view control each time you click the control in the Toolbox.

8. Configure the properties, as required.

**Note:**

To view the full list of properties, select the **Advanced** check box.

9. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Add a tree view item to a tree view item

Before you begin

You must add a tree view and at least one top-level tree view item before you can nest tree view items.

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.

5. Add a tree view control with at least one tree view item.
6. Choose one of the following methods to add a tree view item to the existing tree view item:
 - In the Document Tree, right-click the tree view item you want to add a nested menu item to, and then select **Add Tree View Item**.
 - Right-click the tree view item in the designer, and then select **Add Tree View Item**.
 - Select the tree view item in the designer, and then from the **Items Control** list, click **TreeView Item [add]**. A tree view item appears in the tree view control each time you click the control in the Toolbox.

**Tip:**

Resize the tree view control to view all of the tree view items.

7. Configure the properties, as required.

**Note:**

To view the full list of properties, select the **Advanced** check box.

8. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Combo Box



A Combo Box control consists of a list and a selection field. The list presents the options that you can select, and the selection field displays the selected list item.

If you select the `IsEditable` property check box, you can choose to enter information that is not available in the list to the selection field. If the `IsEditable` property check box is cleared, you can only select from the available items in the list.

You can use the `ItemsSource` property to configure a combo box to get its item list from a data source, or you can manually add the list items and their labels.

See **Items Controls** to view the usage and description of the properties for this control.

Related reference

[Items Controls \(on page 904\)](#)

Combo Item [add]



The Combo Item [add] control is used to add list items to a combo box.

The list items are not visible in design mode. The labels for the individual list items are configured in the Properties panel.

See **Items Controls** to view the usage and description of the properties for this control.

Related reference

[Items Controls \(on page 904\)](#)

Group Box



A Group Box is a WPF control that is a rectangle that surrounds a set of controls, such as check boxes or radio buttons, under a heading.

Group boxes organize controls related by a common purpose. Unlike an expander control, the group box always displays all of its contents, grouped together in an outlined box.

See **Items Controls** to view the usage and description of the properties for this control.

Related reference

[Items Controls \(on page 904\)](#)

Items Control

The Items Control is a Silverlight control that is used to contain another control. Whereas other controls in the Items Control group are used to contain a collection of specific child controls (for example, a list box is used to contain a collection of list box items), the Items Control can contain any control.

Use the ItemsSource property to bind to the control that you want contained within the Items Control.

See **Items Controls** to view the usage and description of the properties for this control.

Related reference[Items Controls \(on page 904\)](#)

List Box



A List Box control displays a collection of selectable items. A list box typically contains list box items; however, it can also contain a list of any of the available controls.

A list box has horizontal and vertical scrolling capabilities. When the number of list items exceeds the list box's defined area, a vertical scroll bar automatically appears. The placement of items within the list box can be adjusted through the box's `FlowDirection` and `VerticalContent Alignment` or `HorizontalContentAlignment` properties.

By default, the `SelectionMode` property is set to `single`, which allows only a single item to be selected from the list box. You can change the `SelectionMode` property to allow an operator to select multiple non-contiguous items or a combination of contiguous and non-contiguous items.

A list box has no `Content` property; the items collection is the content. Changes to list box items can be made by adjusting certain properties of the list box itself. For example, a change to the list box's `FontSize` property cascades through the box's entire list of items.

See **Items Controls** to view the usage and description of the properties for this control.

Related reference[Items Controls \(on page 904\)](#)

List Box Item [add]



The List Box Item [add] control is used to add list items to a list box. The list items are visible in design mode if the list box size is configured to allow this.

The labels for the individual list items are configured in the Properties panel.

See **Items Controls** to view the usage and description of the properties for this control.

Related reference[Items Controls \(on page 904\)](#)

List View



The List View is a WPF control that can be added to a form or user display with list view items added to it.

This control is generally authored through code behind by assigning a grid view to the View property and applying data binding.

See **Items Controls** to view the usage and description of the properties for this control.

Related reference

[Items Controls \(on page 904\)](#)

List View Item [add]



The List View Item [add] is a WPF control that is used to add list items to a list view. The list items are visible in design mode if the list view size is configured to allow this.

The labels for the individual list items are configured in the Properties panel.

See **Items Controls** to view the usage and description of the properties for this control.

Related reference

[Items Controls \(on page 904\)](#)

Menu



The Menu control allows you to create both statically and dynamically displayed menus. You can manually configure the contents of the menu, or you can bind to a data source.

A menu has layout and resizing capabilities for its items collection. When the number of list items exceeds the menu's defined area, the menu control automatically wraps the next list item down to the next line. If the menu itself is resized to the point where its right edge touches the right edge of the list items, its items collection is wrapped and reorganized within the menu's new boundaries.

You can add submenu items to menu items and to other submenu items.

See **Items Controls** to view the usage and description of the properties for this control.

Related reference

[Items Controls \(on page 904\)](#)

Menu Item [add]



The Menu Item [add] control is used to add list items to a menu. The list items are visible in design mode if the menu size is configured to allow this.

The labels for the individual list items are configured in the Properties panel.

See **Items Controls** to view the usage and description of the properties for this control.

Related reference

[Items Controls \(on page 904\)](#)

Separator



A Separator is a control without a Header or Content properties. It cannot accept any child controls. It resembles a horizontal or vertical line (depending on the items control it is placed into), and is used as a simple divider to split or separate items within an items collection into groups.

Separators are commonly used with items controls such as menus/menu items, toolbars, and status bars. Separators are not selectable objects.

See **Items Controls** to view the usage and description of the properties for this control.

Related reference

[Items Controls \(on page 904\)](#)

Status Bar



A Status Bar is a WPF control that is usually located across the bottom of an application window and contains hints, tips, or other information about the status of what is happening inside an active (that is, running) application.

See **Items Controls** to view the usage and description of the properties for this control.

Related reference

[Items Controls \(on page 904\)](#)

Status Bar Item [add]



The Status Bar Item is a WPF control that is combined with a status bar.

The Content property of the status bar item is then set to text or another control.

See **Items Controls** to view the usage and description of the properties for this control.

Related reference

[Items Controls \(on page 904\)](#)

Tab



The Tab control is an items control configured to display its items collection as tabs within a defined area.

When the tab control is resized, its items collection of tabs wraps and resizes accordingly. The control's TabStripPlacement property determines whether the tabs show at the top, bottom, left or right of the control.

See **Items Controls** to view the usage and description of the properties for this control.

Related reference

[Items Controls \(on page 904\)](#)

Tab Item [add]



Tab Item [add] controls are combined with a Tab control. The Header property sets the text or any control to display on the tab of the Tab Item.

The Content property specifies the control to display on the body of the Tab Item.

See **Items Controls** to view the usage and description of the properties for this control.

Related reference

[Items Controls \(on page 904\)](#)

Tool Bar



The Tool Bar is a WPF control that is used as a container for groups of commands or controls that are typically related in their function. A tool bar usually contains buttons that invoke commands.

See **Items Controls** to view the usage and description of the properties for this control.

Related reference

[Items Controls \(on page 904\)](#)

Tool Bar Tray



A Tool Bar Tray is a WPF control that is a container for toolbars. A tool bar tray allows limited repositioning of its child toolbars using drag and drop. Tool bar trays can have multiple rows of toolbars.

The Band property allows you to set a toolbar on a particular row of a tool bar tray.

See **Items Controls** to view the usage and description of the properties for this control.

Related reference

[Items Controls \(on page 904\)](#)

Tree View



A Tree View control organizes its tree view item collection into groups in a hierarchal fashion. While the hierarchal structure is somewhat similar to a menu (for example, the tree view also uses a main item and a list of sub-items to build an organized list of its items collection), the tree view uses a somewhat different layout. Tree view items with connected sub-items are expandable and collapsible.

A tree view initiates horizontal and/or vertical scroll bars if it is resized to the point where its bottom or right-most edge meets the edges of its contents.

See **Items Controls** to view the usage and description of the properties for this control.

Related reference[Items Controls \(on page 904\)](#)

Tree View Item [add]



Tree View Item [add] controls are combined with the tree view control. The Header property contains the title or any control for the title of the node. The tree view items collection can be other tree view items or other controls.

See **Items Controls** to view the usage and description of the properties for this control.

Related reference[Items Controls \(on page 904\)](#)

Label Controls

Label controls are used to display text and/or numeric values for real-time data.

Label Controls Property Usage

The following table shows the properties used by each control:

Property	Data Label	Generic Label	Indicator Label	Variance Label
AllowDrop	X	X	X	X
Background	X	X	X	X
BorderBrush	X	X	X	X
BorderThickness	X	X	X	X
ClipToBounds	X	X	X	X
Column	X	X	X	X
ColumnSpan	X	X	X	X
Content		X	X	X
ContentStringFormat		X	X	X
CornerRadius		X	X	X

Property	Data Label	Generic Label	Indicator Label	Variance Label
Effect	X	X	X	X
FlowDirection	X	X	X	X
Focusable	X	X	X	X
FocusVisualStyle	X	X	X	X
FontFamily	X	X	X	X
FontSize	X	X	X	X
FontStretch	X	X	X	X
FontStyle	X	X	X	X
FontWeight	X	X	X	X
ForceCursor	X	X	X	X
Foreground	X	X	X	X
GlareBrush		X	X	X
HeaderBackground	X			
HeaderCaption	X			
HeaderFontSize	X			
HeaderForeground	X			
Height	X	X	X	X
HorizontalAlign- ment	X	X	X	X
HorizontalContent- Alignment	X	X	X	X
IndicatorAbove- MaxBackground- Brush	X			
IndicatorAbove- MaxGoreground- Brush	X			

Property	Data Label	Generic Label	Indicator Label	Variance Label
IndicatorBelowMin- BackgroundBrush	X			
IndicatorBelowMin- ForegroundBrush	X			
IndicatorMax- Threshold	X			
IndicatorMin- Threshold	X			
IndicatorPulseDe- lay	X			
InnerBorderBrush	X			X
InnerBorderThick- ness	X			X
InnerCornerRadius	X			X
IsEnabled	X	X	X	X
IsHitTestVisible	X	X	X	X
IsManipulationEn- abled	X	X	X	X
IsSharedSizeScope	X	X	X	X
IsTabStop	X	X	X	X
Localization	X	X	X	X
Margin	X	X	X	X
MaxHeight	X	X	X	X
MaxWidth	X	X	X	X
MinHeight	X	X	X	X
Minimum	X			
MinWidth	X	X	X	X
Name	X	X	X	X

Property	Data Label	Generic Label	Indicator Label	Variance Label
Opacity	X	X	X	X
OpacityMask	X	X	X	X
OverridesDefault- Style	X	X	X	X
Padding	X	X	X	X
RenderTransform- Origin	X	X	X	X
Row	X	X	X	X
RowSpan	X	X	X	X
SnapsToDevicePix- els	X	X	X	X
StartValue				X
Style	X	X	X	X
TabIndex	X	X	X	X
Tag	X	X	X	X
Template	X	X	X	X
ToolTip	X	X	X	X
Uid	X	X	X	X
UnitCaption	X			
UnitFontSize	X			
UseLayoutRound- ing	X	X	X	X
Value	X		X	X
ValueFormat				X
VarianceFormat				X
VerticalAlignment	X	X	X	X
VerticalContentAl- ignment	X	X	X	X

Property	Data Label	Generic Label	Indicator Label	Variance Label
Visibility	X	X	X	X
Width	X	X	X	X

Label Controls Property Descriptions

The following table shows the properties descriptions used by all control:

Property	Description
AllowDrop	Select this check box to indicate that the control can accept data that is dragged onto it.
Background	Expand this field to select the background color of the selected control.
BorderBrush	Expand this field to select the border color of the selected control.
BorderThickness	Specifies the border thickness of a control, in pixels.
ClipToBounds	Select this check box to clip the content of this control (or content coming from the child elements of this control) to fit into the size of the containing control.
Column	Specifies the column within the grid where the control is located.
ColumnSpan	Specifies the total number of columns that child content spans within a grid.
Content	Specifies the content displayed on the control.
ContentStringFormat	Specifies a composite string that determines how to format the Content property if it is displayed as a string.
CornerRadius	Specifies the degree to which the corners of a control are rounded.
Effect	Click the arrow to expand the property, select the effect you want to apply to the control, and then

Property	Description
	<p>configure the specific values for the effect. Valid values to select are:</p> <ul style="list-style-type: none"> • Blur <ul style="list-style-type: none"> ◦ Kernel Type ◦ Radius • Drop Shadow <ul style="list-style-type: none"> ◦ Color ◦ Direction ◦ Opactiy ◦ Shadow Depth
FlowDirection	<p>Specifies the direction that text and other user interface (UI) elements flow within any parent control. Selections are:</p> <ul style="list-style-type: none"> • Left to Right • Right to Left
Focusable	<p>Select this check box to indicate that the control can receive focus.</p>
FocusVisualStyle	<p>Specifies a property that enables customization of appearance, effects, or other style characteristics that apply to this control when it has focus.</p>
FontFamily	<p>Specifies the font family for the control.</p>
FontSize	<p>Specifies the font size for the control.</p>
FontStretch	<p>Specifies the degree to which a font is condensed or expanded on the screen.</p>
FontStyle	<p>Specifies the font style for the control.</p>
FontWeight	<p>Specifies the weight or thickness of the specified font.</p>
ForceCursor	<p>Select this check box to indicate that the user interface (UI) renders the cursor as declared by the Cursor property.</p>

Property	Description
Foreground	Expand this field to select the foreground color.
GlareBrush	Expand this field to specify the glare effect for the control.
HeaderBackground	Expand this field to select the header background color.
HeaderCaption	Specifies the header text for the control.
HeaderFontSize	Specifies the font size for the text in the header.
HeaderForeground	Expand this field to select the header text color.
Height	Specifies the height of the control, in pixels.
HorizontalAlignment	<p>Specifies the horizontal alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
HorizontalContentAlignment	<p>Specifies the horizontal alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
ImageStretch	<p>Specifies how the image should be stretched to fill the destination rectangle. Selections are:</p> <ul style="list-style-type: none"> • Fill • None • Uniform • UniformToFill

Property	Description
IndicatorAboveMaxBackgroundBrush	Expand this field to select the color to use for the background of the label when the value is greater than the IndicatorMaxThreshold value.
IndicatorAboveMaxForegroundBrush	Expand this field to select the color to use for the contents of the label when the value is greater than the IndicatorMaxThreshold value.
IndicatorBelowMinBackgroundBrush	Expand this field to select the color to use for the background of the label when the value is less than the IndicatorMaxThreshold value.
IndicatorBelowMinForegroundBrush	Expand this field to select the color to use for the contents of the label when the value is less than the IndicatorMaxThreshold value.
IndicatorMaxThreshold	Specifies the value at which the high-level indicator is triggered.
IndicatorMinThreshold	Specifies the value at which the low-level indicator is triggered.
IndicatorPulseDelay	Specifies the number of milliseconds between flashes when the indicator is flashing.
InnerBorderBrush	Expand this field to select the color for the inner border.
InnerBorderThickness	Expand this field to select the color for the inner border.
InnerCornerRadius	Specifies the degree to which the corners of the inner border are rounded.
IsEnabled	Select this check box to indicate that this control is enabled in the user interface (UI).
IsHitTestVisible	Select this check box to indicate that the control can possibly be returned as a hit test result from some portion of its rendered content.
IsManipulationEnabled	Select this check box to indicate that the manipulation events for the control are enabled.

Property	Description
IsSharedSizeScope	Select this check box to indicate that controls are sharing size information.
IsTabStop	Select this check box to indicate that a control is included in tab navigation.
Localization	Click the ellipsis button to open the Localization Configuration dialog box where you can configure the property to localize, the language to use for localization, and the text to localize.
Margin	Specifies the outer margin of a control.
MaxHeight	Specifies the maximum height of a control.
MaxWidth	Specifies the maximum width of a control.
MinHeight	Specifies the minimum height of a control.
Minimum	Specifies the minimum value of the range of the control.
MinWidth	Specifies the minimum width of a control.
Name	Specifies the identifying name of the control. The name provides a reference so that code behind, such as event handler code, can refer to a markup element during processing by a XAML processor.
Opacity	Specifies the opacity factor applied to the entire control. Valid values are 0 through 1, with up to 15 decimal places.
OpacityMask	Expand this field to select a color for the opacity mask, which is applied to any alpha-channel masking for the rendered content of this control.
OverridesDefaultStyle	Select this check box to indicate that this control incorporates style properties from theme styles.
Padding	Specifies the amount of padding to apply to the inside edge of each side of a control.

Property	Description
RenderTransformOrigin	<p>Specifies the center point of any possible render transform, relative to the bounds of the element.</p> <ul style="list-style-type: none"> • X is the horizontal origin factor, given as a value between 0 and 1. • Y is the vertical origin factor, given as a value between 0 and 1.
Row	RowBackground
RowSpan	Specifies the total number of rows that child content spans within a grid.
SnapsToDevicePixels	Select this check box to indicate that rendering for this control uses device-specific pixel settings.
StartValue	Specifies the value that the control will compare against when the control is first used.
Style	Specifies the style used by this control when it is rendered.
TabIndex	Specifies the order in which controls receive focus when the Tab key is pressed.
Tag	Specifies an arbitrary object value that can be used to store custom information about this control.
Template	Click the ellipsis button to open the Open dialog box to select a template to use.
ToolTip	Specifies the tool-tip that displays for this control in the user interface (UI).
Uid	Specifies the unique identifier (for localization) for this control.
UnitCaption	Specifies the caption for the units value.
UnitFontSize	Specifies the font size for the units display values.
UseLayoutRounding	Specifies a value that determines whether rendering for the object and its visual subtree should use

Property	Description
	rounding behavior that aligns rendering to whole pixels.
Value	Specifies the current magnitude of the range control.
ValueFormat	Specifies the format the value displays in.
VarianceFormat	Specifies the format the variance value displays in.
VerticalAlignment	<p>Specifies the vertical alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
VerticalContentAlignment	<p>Specifies the vertical alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
Visibility	<p>Specifies the visibility of this control. Selections are:</p> <ul style="list-style-type: none"> • Collapsed • Hidden • Visible
Width	Specifies the width of the control, in pixels.

Data Label



The Data Label control is a text label that displays data—both text and numerical—for a specified point or trendline in a chart.

See **Label Controls** to view the usage and description of the properties for this control.

Related reference

[Label Controls \(on page 945\)](#)

Generic Label



The Generic Label control is a standard text box control that displays text.

See **Label Controls** to view the usage and description of the properties for this control.

Related reference

[Label Controls \(on page 945\)](#)

Indicator Label



The Indicator Label control is a text label that changes background and foreground colors depending on the indicator value. The state is determined by the value assigned to the control.

See **Label Controls** to view the usage and description of the properties for this control.

Related reference

[Label Controls \(on page 945\)](#)

Variance Label



The Variance Label control displays the current value given by real-time data. It also caches and displays the last value with either a green upward pointing arrow or a red downward pointing arrow to display either an increase or decrease in value since the last update.

The VarianceFormat property is used to set the number variables.

See **Label Controls** to view the usage and description of the properties for this control.

Related reference

[Label Controls \(on page 945\)](#)

Language Switching

The Designer provides a language switching control that allows you to change the text in a form or user display from one language to another during run time.



Note:

Language switching is available in both the WPF Designer and the Silverlight Designer.

Language Switching Controls Property Usage

The following table shows the properties used by each control:

Property	Flag Switching Button	Switch Language Button
AllowDrop	X	X
Background	X	X
BorderBrush	X	X
BorderThickness	X	X
ClickMode	X	X
ClipToBounds	X	X
Column	X	X
ColumnSpan	X	X
CommandParameter	X	X
CommandTarget	X	X
Content	X	X
ContentStringFormat	X	X
DefaultFlag	X	
Effect	X	X
FlagsParameters	X	
FlagStyle	X	
FlowDirection	X	X
Focusable	X	X

Property	Flag Switching Button	Switch Language Button
FocusBrush	X	
FocusVisualStyle	X	X
FontFamily	X	X
FontSize	X	X
FontStretch	X	X
FontStyle	X	X
FontWeight	X	X
ForceCursor	X	X
Foreground	X	X
Height	X	X
HorizontalAlignment	X	X
HorizontalContentAlignment	X	X
IsCancel	X	X
IsDefault	X	X
IsEnabled	X	X
IsHitTestVisible	X	X
IsManipulationEnabled	X	X
IsSharedSizeScope	X	X
IsTabStop	X	X
Localization	X	X
Margin	X	X
MaxHeight	X	X
MaxWidth	X	X
MinHeight	X	X
MinWidth	X	X
Name	X	X

Property	Flag Switching Button	Switch Language Button
Opacity	X	X
OpacityMask	X	X
OverridesDefaultStyle	X	X
Padding	X	X
RenderTransformOrigin	X	X
Row	X	X
RowSpan	X	X
SnapsToDevicePixels	X	X
Style	X	X
TabIndex	X	X
Tag	X	X
Template	X	X
ToolTip	X	X
Uid	X	X
UseLayoutRounding	X	X
VerticalAlignment	X	X
VerticalContentAlignment	X	X
Visibility	X	X
Width	X	X

Language Switching Controls Property Descriptions

The following table shows the properties descriptions used by all control:

Property	Description
AllowDrop	Select this check box to indicate that the control can accept data that is dragged onto it.
Background	Expand this field to select the background color of the selected control.

Property	Description
BorderBrush	Expand this field to select the border color of the selected control.
BorderThickness	Specifies the border thickness of a control, in pixels.
ClickMode	<p>Specifies when the Click event occurs. Selections are:</p> <ul style="list-style-type: none"> • Hover • Press • Release
ClipToBounds	Select this check box to clip the content of this control (or content coming from the child elements of this control) to fit into the size of the containing control.
Column	Specifies the column within the grid where the control is located.
ColumnSpan	Specifies the total number of columns that child content spans within a grid.
CommandParameter	Specifies the parameter to pass to the Command property.
CommandTarget	Specifies the element on which to raise the specified command.
Content	Specifies the content displayed on the control.
ContentStringFormat	Specifies a composite string that determines how to format the Content property if it is displayed as a string.
DefaultFlag	<p>Specifies the default flag to use for the flag switching button. This is the flag that is initially displayed at run time. Valid selections are:</p> <ul style="list-style-type: none"> • Chinese • English

Property	Description
	<ul style="list-style-type: none"> • French • German • Italian • Korean • Polish • Russian • Spanish
Effect	<p>Click the arrow to expand the property, select the effect you want to apply to the control, and then configure the specific values for the effect. Valid values to select are:</p> <ul style="list-style-type: none"> • Blur <ul style="list-style-type: none"> ◦ Kernel Type ◦ Radius • Drop Shadow <ul style="list-style-type: none"> ◦ Color ◦ Direction ◦ Opactiy ◦ Shadow Depth
FlagsParameters	<p>Click the ellipsis button to open the Language Selection dialog box, where you can select the flags you want to be able to switch between. Valid selections are:</p> <ul style="list-style-type: none"> • Chinese • English • French • German • Italian • Korean • Polish • Russian • Spanish

Property	Description
FlagStyle	<p>Specifies the style of the flag control. Valid selections are:</p> <ul style="list-style-type: none"> • Wave • Box
FlowDirection	<p>Specifies the direction that text and other user interface (UI) elements flow within any parent control. Selections are:</p> <ul style="list-style-type: none"> • Left to Right • Right to Left
Focusable	<p>Select this check box to indicate that the control can receive focus.</p>
FocusBrush	<p>Expand this field to select the color that displays around the control's border when the control has focus.</p>
FocusVisualStyle	<p>Specifies a property that enables customization of appearance, effects, or other style characteristics that apply to this control when it has focus.</p>
FontFamily	<p>Specifies the font family for the control.</p>
FontSize	<p>Specifies the font size for the control.</p>
FontStretch	<p>Specifies the degree to which a font is condensed or expanded on the screen.</p>
FontStyle	<p>Specifies the font style for the control.</p>
FontWeight	<p>Specifies the weight or thickness of the specified font.</p>
ForceCursor	<p>Select this check box to indicate that the user interface (UI) renders the cursor as declared by the Cursor property.</p>
Foreground	<p>Expand this field to select the foreground color.</p>
Height	<p>Specifies the height of the control, in pixels.</p>

Property	Description
HorizontalAlignment	<p>Specifies the horizontal alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
HorizontalContentAlignment	<p>Specifies the horizontal alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
IsCancel	<p>Select this check box to indicate that a Button or Repeat Button acts as a Cancel button. Press the Esc key to select the actual Cancel button.</p>
IsDefault	<p>Select this check box to indicate that a Button or Repeat Button is the default Submit button. Press the Enter key to select the actual Submit button.</p>
IsEnabled	<p>Select this check box to indicate that this control is enabled in the user interface (UI).</p>
IsHitTestVisible	<p>Select this check box to indicate that the control can possibly be returned as a hit test result from some portion of its rendered content.</p>
IsManipulationEnabled	<p>Select this check box to indicate that the manipulation events for the control are enabled.</p>
IsSharedSizeScope	<p>Select this check box to indicate that controls are sharing size information.</p>
IsTabStop	<p>Select this check box to indicate that a control is included in tab navigation.</p>
Localization	<p>Click the ellipsis button to open the Localization Configuration dialog box where you can configure</p>

Property	Description
	the property to localize, the language to use for localization, and the text to localize.
Margin	Specifies the outer margin of a control.
MaxHeight	Specifies the maximum height of a control.
MaxWidth	Specifies the maximum width of a control.
MinHeight	Specifies the minimum height of a control.
MinWidth	Specifies the minimum width of a control.
Name	Specifies the identifying name of the control. The name provides a reference so that code behind, such as event handler code, can refer to a markup element during processing by a XAML processor.
Opacity	Specifies the opacity factor applied to the entire control. Valid values are 0 through 1, with up to 15 decimal places.
OpacityMask	Expand this field to select a color for the opacity mask, which is applied to any alpha-channel masking for the rendered content of this control.
OverridesDefaultStyle	Select this check box to indicate that this control incorporates style properties from theme styles.
Padding	Specifies the amount of padding to apply to the inside edge of each side of a control.
RenderTransformOrigin	<p>Specifies the center point of any possible render transform, relative to the bounds of the element.</p> <ul style="list-style-type: none"> • X is the horizontal origin factor, given as a value between 0 and 1. • Y is the vertical origin factor, given as a value between 0 and 1.
Row	RowBackground
RowSpan	Specifies the total number of rows that child content spans within a grid.

Property	Description
SnapsToDevicePixels	Select this check box to indicate that rendering for this control uses device-specific pixel settings.
Style	Specifies the style used by this control when it is rendered.
TabIndex	Specifies the order in which controls receive focus when the Tab key is pressed.
Tag	Specifies an arbitrary object value that can be used to store custom information about this control.
Template	Click the ellipsis button to open the Open dialog box to select a template to use.
ToolTip	Specifies the tool-tip that displays for this control in the user interface (UI).
Uid	Specifies the unique identifier (for localization) for this control.
UseLayoutRounding	Specifies a value that determines whether rendering for the object and its visual subtree should use rounding behavior that aligns rendering to whole pixels.
VerticalAlignment	<p>Specifies the vertical alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
VerticalContentAlignment	<p>Specifies the vertical alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch

Property	Description
Visibility	Specifies the visibility of this control. Selections are: <ul style="list-style-type: none"> • Collapsed • Hidden • Visible
Width	Specifies the width of the control, in pixels.

Configure controls to change displayed text between languages

Use the Switch Language Button control to create separate buttons that allow you to change text between different languages. Use the Flag Switching Button control to create a single button that allows you to toggle between languages.

Before you begin



Important:

- You must provide the translated text for the selected properties. The Designer does not automatically translate text.
- The Switch Language button does not act as a toggle button; that is, you cannot use the same button to switch back to the original language. You must add a separate language button for each language that you want to be able to switch to. However, the Flag Switching button does act as a toggle button; that is, you use only the one button to switch between the configured languages.

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. To configure the Switch Language Button control, do the following:
 - a. Expand the **Language Switching** group, and then select the **SwitchLanguageButton** control.
 - b. Click and drag the control to the desired position in the Designer, and then click the **Properties** tab.
 - c. Configure the base properties, as required.

**Note:**

To view the full list of properties, select the Advanced check box.

- d. In the **Content** field, enter the text you want displayed on the language button. For example, if the button will be used to switch text to Spanish, enter `Spanish Or Español`.
 - e. From the **Language ID** list, select the language that the text on the control(s) will be switched to.
6. Repeat steps 4-5 for each separate language button.
 7. To configure the Flag Switching Button control, do the following:
 - a. From the **DefaultFlag** drop-down list, select the language for the flag that you want displayed when the form opens.
 - b. Click the **FlagsParameters** ellipsis button.
The **Language Selection** dialog box appears.
 - c. Select the languages that the text on the control(s) will be switched to, and then click **OK**.
 - d. From the **FlagStyle** drop-down list, select the style of flag you want to use.
 8. If required, add controls whose text you want to switch to an alternate language.
 9. Select a control, and then click the **Properties** tab.
 10. Click the **Localization** ellipsis button.
The **Localization Configuration** dialog box appears.
 11. From the **Property Name** list, select the property whose text you want to switch languages for.
 12. From the **Language** list, select the language you want to switch the text to.

**Note:**

This language must match the **Language ID** property of the switch language button.

13. In the **Text** field, enter the translated text for the selected property, and then click **Apply**.
The information for language switching appears in the **Translations** area.
14. Click **OK**.

**Tip:**

To remove a translation string, select it, click **Delete**, and then click **OK**.

15. Repeat steps 8-13 for each additional control.
16. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Related reference

[Language Switching \(on page 957\)](#)

Related information

[Designer: Control Toolbox \(on page 725\)](#)

Localization Configuration

The Localization property ellipsis button, found on the Properties tab for each control, opens the Localization Configuration dialog box, where you specify which properties you want to configure language switching for, the language to switch to, and the translated text. If you want all of the text for a specific control to switch to another language, you must configure the translated text for each property.



Important:

You must provide the translated text for the selected properties. The Designer does not automatically translate text.

The following table describes the available options.

Attribute	Value	Description
Property Name	System-defined	Specifies the property whose text you want to switch languages for.
Language	System-defined	Specifies the language to switch to.
Text	User-defined	Specifies the translated text to display when the language button is clicked in run time.

Attribute	Value	Description
Apply	N/A	Click this button to add the information to the Translations area.
Translations	User-defined	Displays the property, language, and text to switch to when the language button is clicked during run time.
Delete	N/A	Click this button to delete the selected information from the Translations area.

Flag Switching Button



The Flag Switching Button is a WPF control that is used to dynamically change a control's displayed text from one language to another.

See **Language Switching Controls** to view the usage and description of the properties for this control.

Related reference

[Language Switching \(on page 957\)](#)

Switch Language Button



The Switch Language Button is a WPF control that is used to dynamically change the text displayed on controls from one language to another.

For example, if a form is authored in English and you want an operator to be able to switch the displayed language to Spanish, you can add a language button to the form that automatically changes the text to Spanish when it is clicked.



Important:

The Switch Language button does not act as a toggle button; that is, you cannot use the same button to switch back to the original language. You must add a separate language button for each language that you want to be able to switch to. However, the Flag Switching button does act as a toggle button; that is, you use only the one button to switch between the configured languages.

The Language ID property for the Switch Language Button control is used to specify the language that you want the button to switch the text to.

Language switching is based on specific properties for individual controls. For example, you can affect the Text and ToolTip properties for the text box control, or the CommandParameter, Content, ContentStringFormat, and ToolTip properties for the button control.

See **Language Switching Controls** to view the usage and description of the properties for this control.

Related reference

[Language Switching \(on page 957\)](#)

LED Controls

LED controls have on, off, and pulse abilities and, when grouped, can display alphabetical, numerical, or symbolic characters.

LED Controls Property Usage

The following table shows the properties used by each control:

Property	LED Five by Seven Matrix	LED Light	LED Seven Segment	LED Sixteen Segment
AllowDrop	X	X	X	X
Background	X	X	X	X
BorderBrush	X	X	X	X
BorderThickness	X	X	X	X
ClipToBounds	X	X	X	X
Column	X	X	X	X
ColumnSpan	X	X	X	X
CornerRadius	X	X	X	X
Effect	X	X	X	X
FlowDirection	X	X	X	X
Focusable	X	X	X	X
FocusVisualStyle	X	X	X	X

Property	LED Five by Seven Matrix	LED Light	LED Seven Segment	LED Sixteen Segment
FontFamily	X	X	X	X
FontSize	X	X	X	X
FontStretch	X	X	X	X
FontStyle	X	X	X	X
FontWeight	X	X	X	X
ForceCursor	X	X	X	X
Foreground	X	X	X	X
GlareBrush	X	X	X	X
Height	X	X	X	X
HorizontalAlignment	X	X	X	X
HorizontalContentAlignment	X	X	X	X
InnerBorderBrush	X	X	X	X
InnerBorderThickness	X	X	X	X
InnerCornerRadius	X	X	X	X
IsEnabled	X	X	X	X
IsHitTestVisible	X	X	X	X
IsManipulationEnabled	X	X	X	X
IsSharedSizeScope	X	X	X	X
IsTabStop	X	X	X	X
LEDBrightness	X	X	X	X
LEDBrush	X	X	X	X
LEDDisplayCount	X		X	X
LEDDisplayValue	X	X	X	X

Property	LED Five by Seven Matrix	LED Light	LED Seven Segment	LED Sixteen Segment
LEDIsOn	X	X	X	X
LEDOffOpacity	X	X	X	X
LEDPadding	X	X	X	X
LEDPulse	X	X	X	X
LEDPulseDelay	X	X	X	X
Localization	X	X	X	X
Margin	X	X	X	X
MaxHeight	X	X	X	X
MaxWidth	X	X	X	X
MinHeight	X	X	X	X
MinorBackground	X	X	X	X
MinWidth	X	X	X	X
Name	X	X	X	X
Opacity	X	X	X	X
OpacityMask	X	X	X	X
OverridesDefault-Style	X	X	X	X
Padding	X	X	X	X
RenderTransform-Origin	X	X	X	X
Row	X	X	X	X
RowSpan	X	X	X	X
SnapsToDevicePixels	X	X	X	X
StrokeThickness	X	X	X	X
Style	X	X	X	X

Property	LED Five by Seven Matrix	LED Light	LED Seven Segment	LED Sixteen Segment
TabIndex	X	X	X	X
Tag	X	X	X	X
Template	X	X	X	X
TertiaryBackground	X	X	X	X
TertiaryBorderBrush	X	X	X	X
TertiaryCornerRadius	X	X	X	X
ToolTip	X	X	X	X
Uid	X	X	X	X
UseLayoutRounding	X	X	X	X
VerticalAlignment	X	X	X	X
VerticalContentAlignment	X	X	X	X
Visibility	X	X	X	X
Width	X	X	X	X

LED Controls Property Descriptions

The following table shows the properties descriptions used by all control:

Property	Description
AllowDrop	Select this check box to indicate that the control can accept data that is dragged onto it.
Background	Expand this field to select the background color of the selected control.
BorderBrush	Expand this field to select the border color of the selected control.

Property	Description
BorderThickness	Specifies the border thickness of a control, in pixels.
ClipToBounds	Select this check box to clip the content of this control (or content coming from the child elements of this control) to fit into the size of the containing control.
Column	Specifies the column within the grid where the control is located.
ColumnSpan	Specifies the total number of columns that child content spans within a grid.
CornerRadius	Specifies the degree to which the corners of a control are rounded.
Effect	<p>Click the arrow to expand the property, select the effect you want to apply to the control, and then configure the specific values for the effect. Valid values to select are:</p> <ul style="list-style-type: none"> • Blur <ul style="list-style-type: none"> ◦ Kernel Type ◦ Radius • Drop Shadow <ul style="list-style-type: none"> ◦ Color ◦ Direction ◦ Opacity ◦ Shadow Depth
FlowDirection	<p>Specifies the direction that text and other user interface (UI) elements flow within any parent control. Selections are:</p> <ul style="list-style-type: none"> • Left to Right • Right to Left
Focusable	Select this check box to indicate that the control can receive focus.

Property	Description
FocusVisualStyle	Specifies a property that enables customization of appearance, effects, or other style characteristics that apply to this control when it has focus.
FontFamily	Specifies the font family for the control.
FontSize	Specifies the font size for the control.
FontStretch	Specifies the degree to which a font is condensed or expanded on the screen.
FontStyle	Specifies the font style for the control.
FontWeight	Specifies the weight or thickness of the specified font.
ForceCursor	Select this check box to indicate that the user interface (UI) renders the cursor as declared by the Cursor property.
Foreground	Expand this field to select the foreground color.
GlareBrush	Expand this field to specify the glare effect for the control.
Height	Specifies the height of the control, in pixels.
HorizontalAlignment	<p>Specifies the horizontal alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
HorizontalContentAlignment	<p>Specifies the horizontal alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch

Property	Description
ImageStretch	<p>Specifies how the image should be stretched to fill the destination rectangle. Selections are:</p> <ul style="list-style-type: none"> • Fill • None • Uniform • UniformToFill
InnerBorderBrush	Expand this field to select the color for the inner border.
InnerBorderThickness	Expand this field to select the color for the inner border.
InnerCornerRadius	Specifies the degree to which the corners of the inner border are rounded.
IsEnabled	Select this check box to indicate that this control is enabled in the user interface (UI).
IsHitTestVisible	Select this check box to indicate that the control can possibly be returned as a hit test result from some portion of its rendered content.
IsManipulationEnabled	Select this check box to indicate that the manipulation events for the control are enabled.
IsSharedSizeScope	Select this check box to indicate that controls are sharing size information.
IsTabStop	Select this check box to indicate that a control is included in tab navigation.
LEDBrightness	Specifies the opacity level on the control, arranged from 0 to 1.
LEDBrush	Expand this field to select the color for the LED background.
LEDDisplayCount	Specifies the number of LED segments to display.
LEDDisplayValue	Specifies the alphanumeric value on the LED display.

Property	Description
LEDIsOn	Select this check box to indicate that the LED control is on.
LEDOffOpacity	Specifies the opacity value on the LED display when it is in the Off state.
LEDPadding	Specifies the horizontal and vertical padding within the LED control's parent.
LEDPulse	Select this check box to indicate that the LED control will pulse.
LEDPulseDelay	Specifies the flash interval for the pulse.
Localization	Click the ellipsis button to open the Localization Configuration dialog box where you can configure the property to localize, the language to use for localization, and the text to localize.
Margin	Specifies the outer margin of a control.
MaxHeight	Specifies the maximum height of a control.
MaxWidth	Specifies the maximum width of a control.
MinHeight	Specifies the minimum height of a control.
MinorBackground	Expand this field to select the fill color of secondary components on the control.
MinWidth	Specifies the minimum width of a control.
Name	Specifies the identifying name of the control. The name provides a reference so that code behind, such as event handler code, can refer to a markup element during processing by a XAML processor.
Opacity	Specifies the opacity factor applied to the entire control. Valid values are 0 through 1, with up to 15 decimal places.
OpacityMask	Expand this field to select a color for the opacity mask, which is applied to any alpha-channel masking for the rendered content of this control.

Property	Description
OverridesDefaultStyle	Select this check box to indicate that this control incorporates style properties from theme styles.
Padding	Specifies the amount of padding to apply to the inside edge of each side of a control.
RenderTransformOrigin	<p>Specifies the center point of any possible render transform, relative to the bounds of the element.</p> <ul style="list-style-type: none"> • X is the horizontal origin factor, given as a value between 0 and 1. • Y is the vertical origin factor, given as a value between 0 and 1.
Row	RowBackground
RowSpan	Specifies the total number of rows that child content spans within a grid.
SnapsToDevicePixels	Select this check box to indicate that rendering for this control uses device-specific pixel settings.
StrokeThickness	Specifies the width of the control's outline.
Style	Specifies the style used by this control when it is rendered.
TabIndex	Specifies the order in which controls receive focus when the Tab key is pressed.
Tag	Specifies an arbitrary object value that can be used to store custom information about this control.
Template	Click the ellipsis button to open the Open dialog box to select a template to use.
TertiaryBackground	Expand this field to select the inner background color.
TertiaryBorderBrush	Expand this field to select the innermost border color of the selected control.
TertiaryCornerRadius	Specifies the degree to which the innermost corners of a control are rounded.

Property	Description
ToolTip	Specifies the tool-tip that displays for this control in the user interface (UI).
Uid	Specifies the unique identifier (for localization) for this control.
UseLayoutRounding	Specifies a value that determines whether rendering for the object and its visual subtree should use rounding behavior that aligns rendering to whole pixels.
VerticalAlignment	<p>Specifies the vertical alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
VerticalContentAlignment	<p>Specifies the vertical alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
Visibility	<p>Specifies the visibility of this control. Selections are:</p> <ul style="list-style-type: none"> • Collapsed • Hidden • Visible
Width	Specifies the width of the control, in pixels.

LED Five by Seven Matrix



The LED Panel Five by Seven Matrix control is a panel of five-by-seven matrix LEDs. This panel is used to display alphanumeric data.

See **LED Controls** to view the usage and description of the properties for this control.

Related reference

[LED Controls \(on page 970\)](#)

LED Light



The LED Light control is a single LED that displays in on, off, or pulse states. The LEDBrush property is used to change the color of the LED light.

See **LED Controls** to view the usage and description of the properties for this control.

Related reference

[LED Controls \(on page 970\)](#)

LED Seven Segment



The LED Panel Seven Segment control is a panel of seven segment LEDs. This panel is used to display numeric data.

See **LED Controls** to view the usage and description of the properties for this control.

Related reference

[LED Controls \(on page 970\)](#)

LED Sixteen Segment



The LED Panel Sixteen Segment control is a panel of sixteen segment LEDs. This panel is used to display alphanumeric data.

See **LED Controls** to view the usage and description of the properties for this control.

Related reference

[LED Controls \(on page 970\)](#)

Media Controls

The media of a communications session is the form in which the data is transmitted. Media controls allow an application to recognize a variety of media types and adjust aspects of the media stream, such as the volume of voice transmission.

Media Controls Property Usage

The following table shows the properties used by each control:

Property	Media Control	Video Text Block
AllowDrop	X	X
Background	X	X
BorderBrush	X	X
BorderThickness	X	X
ClipToBounds	X	X
Column	X	X
ColumnSpan	X	X
CornerRadius		X
Effect	X	X
FlowDirection	X	X
Focusable	X	X
FocusVisualStyle	X	X
FontFamily	X	X
FontSize	X	X
FontStretch	X	X
FontStyle	X	X
FontWeight	X	X
ForceCursor	X	X
Foreground	X	X
GlareBrush		

Property	Media Control	Video Text Block
Height	X	X
HorizontalAlignment	X	X
HorizontalContentAlignment	X	X
IsEnabled	X	X
IsHitTestVisible	X	X
IsManipulationEnabled	X	X
IsPlaying	X	X
IsSharedSizeScope	X	X
IsTabStop	X	X
Localization	X	X
Margin	X	X
MaxHeight	X	X
MaxWidth	X	X
MinHeight	X	X
MinWidth	X	X
Name	X	X
Opacity	X	X
OpacityMask	X	X
OverridesDefaultStyle	X	X
Padding	X	X
Position	X	X
RadiusX	X	X
RadiusY	X	X
RenderTransformOrigin	X	X
Repeat		X
Row	X	X

Property	Media Control	Video Text Block
RowSpan	X	X
SnapsToDevicePixels	X	X
Source	X	X
StrokeThickness	X	X
Style	X	X
TabIndex	X	X
Tag	X	X
Template	X	X
Text		X
ToolTip	X	X
Uid	X	X
UseLayoutRounding	X	X
VerticalAlignment	X	X
VerticalContentAlignment	X	X
Visibility	X	X
Width	X	X

Media Controls Property Descriptions

The following table shows the properties descriptions used by all control:

Property	Description
AllowDrop	Select this check box to indicate that the control can accept data that is dragged onto it.
Background	Expand this field to select the background color of the selected control.
BorderBrush	Expand this field to select the border color of the selected control.

Property	Description
BorderThickness	Specifies the border thickness of a control, in pixels.
ClipToBounds	Select this check box to clip the content of this control (or content coming from the child elements of this control) to fit into the size of the containing control.
Column	Specifies the column within the grid where the control is located.
ColumnSpan	Specifies the total number of columns that child content spans within a grid.
CornerRadius	Specifies the degree to which the corners of a control are rounded.
Effect	<p>Click the arrow to expand the property, select the effect you want to apply to the control, and then configure the specific values for the effect. Valid values to select are:</p> <ul style="list-style-type: none"> • Blur <ul style="list-style-type: none"> ◦ Kernel Type ◦ Radius • Drop Shadow <ul style="list-style-type: none"> ◦ Color ◦ Direction ◦ Opactiy ◦ Shadow Depth
FlowDirection	<p>Specifies the direction that text and other user interface (UI) elements flow within any parent control. Selections are:</p> <ul style="list-style-type: none"> • Left to Right • Right to Left
Focusable	Select this check box to indicate that the control can receive focus.

Property	Description
FocusVisualStyle	Specifies a property that enables customization of appearance, effects, or other style characteristics that apply to this control when it has focus.
FontFamily	Specifies the font family for the control.
FontSize	Specifies the font size for the control.
FontStretch	Specifies the degree to which a font is condensed or expanded on the screen.
FontStyle	Specifies the font style for the control.
FontWeight	Specifies the weight or thickness of the specified font.
ForceCursor	Select this check box to indicate that the user interface (UI) renders the cursor as declared by the Cursor property.
Foreground	Expand this field to select the foreground color.
GlareBrush	Expand this field to specify the glare effect for the control.
Height	Specifies the height of the control, in pixels.
HorizontalAlignment	<p>Specifies the horizontal alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
HorizontalContentAlignment	<p>Specifies the horizontal alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch

Property	Description
ImageStretch	<p>Specifies how the image should be stretched to fill the destination rectangle. Selections are:</p> <ul style="list-style-type: none"> • Fill • None • Uniform • UniformToFill
IsEnabled	Select this check box to indicate that this control is enabled in the user interface (UI).
IsHitTestVisible	Select this check box to indicate that the control can possibly be returned as a hit test result from some portion of its rendered content.
IsManipulationEnabled	Select this check box to indicate that the manipulation events for the control are enabled.
IsPlaying	Select this check box to indicate that the control is playing.
IsSharedSizeScope	Select this check box to indicate that controls are sharing size information.
IsTabStop	Select this check box to indicate that a control is included in tab navigation.
Localization	Click the ellipsis button to open the Localization Configuration dialog box where you can configure the property to localize, the language to use for localization, and the text to localize.
Margin	Specifies the outer margin of a control.
MaxHeight	Specifies the maximum height of a control.
MaxWidth	Specifies the maximum width of a control.
MinHeight	Specifies the minimum height of a control.
MinWidth	Specifies the minimum width of a control.
Name	Specifies the identifying name of the control. The name provides a reference so that code behind,

Property	Description
	such as event handler code, can refer to a markup element during processing by a XAML processor.
Opacity	Specifies the opacity factor applied to the entire control. Valid values are 0 through 1, with up to 15 decimal places.
OpacityMask	Expand this field to select a color for the opacity mask, which is applied to any alpha-channel masking for the rendered content of this control.
OverridesDefaultStyle	Select this check box to indicate that this control incorporates style properties from theme styles.
Padding	Specifies the amount of padding to apply to the inside edge of each side of a control.
Position	Specifies the current position of progress through the media's playback time.
RadiusX	Specifies the x-axis radius of the ellipse that is used to round the corners of the rectangle.
RadiusY	Specifies the y-axis radius of the ellipse that is used to round the corners of the rectangle.
RenderTransformOrigin	<p>Specifies the center point of any possible render transform, relative to the bounds of the element.</p> <ul style="list-style-type: none"> • X is the horizontal origin factor, given as a value between 0 and 1. • Y is the vertical origin factor, given as a value between 0 and 1.
Repeat	Select this check box to indicate that the video will repeat after the media has finished running through a continuous video cycle.
Row	RowBackground
RowSpan	Specifies the total number of rows that child content spans within a grid.

Property	Description
SnapsToDevicePixels	Select this check box to indicate that rendering for this control uses device-specific pixel settings.
Source	<p>Click the ellipsis button to open the Navigate Uri dialog box to select the image source to use for the control.</p> <ul style="list-style-type: none"> • Select the From Resource option to select an image source from the Resources directory in the Document Tree panel. • Select the From URI option to select the image source from a remote location.
StrokeThickness	Specifies the width of the control's outline.
Style	Specifies the style used by this control when it is rendered.
TabIndex	Specifies the order in which controls receive focus when the Tab key is pressed.
Tag	Specifies an arbitrary object value that can be used to store custom information about this control.
Template	Click the ellipsis button to open the Open dialog box to select a template to use.
Text	Specifies the text contents of a control.
ToolTip	Specifies the tool-tip that displays for this control in the user interface (UI).
Uid	Specifies the unique identifier (for localization) for this control.
UseLayoutRounding	Specifies a value that determines whether rendering for the object and its visual subtree should use rounding behavior that aligns rendering to whole pixels.
VerticalAlignment	Specifies the vertical alignment characteristics applied to this control. Selections are:

Property	Description
	<ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
VerticalContentAlignment	<p>Specifies the vertical alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
Visibility	<p>Specifies the visibility of this control. Selections are:</p> <ul style="list-style-type: none"> • Collapsed • Hidden • Visible
Width	Specifies the width of the control, in pixels.

Media Control



The Media Control is a control that plays video or audio and extends the media element component in .NET allowing you more flexibility, such as specifying a background color or adding a RadiusX and/or RadiusY to the control for more interesting and dynamic visual effects. The media control can also be used to create digital signage content.

See **Media Controls** to view the usage and description of the properties for this control.

Related reference

[Media Controls \(on page 981\)](#)

Video Text Block



The Video Text Block control is derived from the Media Control but will repeat the video after the media has finished giving the viewer a continuous video cycle. The video text block adds an attention-grabbing effect to any digital signage content.

See **Media Controls** to view the usage and description of the properties for this control.

Related reference

[Media Controls \(on page 981\)](#)

Miscellaneous Controls

The Miscellaneous controls group contains controls that would otherwise be ungrouped.

Miscellaneous Controls Property Usage

The following table shows the properties used by each control:

Property	Animated Fire
AllowDrop	X
Background	X
BorderBrush	X
BorderThickness	X
ClipToBounds	X
Column	X
ColumnSpan	X
Effect	X
FlamelsOn	X
FlameMaxHeight	X
FlameMinHeight	X
FlickerFadeSpeec	X
FlickerSpeed	X
FlowDirection	X

Property	Animated Fire
Focusable	X
FocusVisualStyle	X
FontFamily	X
FontSize	X
FontStretch	X
FontStyle	X
FontWeight	X
ForceCursor	X
Foreground	X
Height	X
HorizontalAlignment	X
HorizontalContentAlignment	X
IsEnabled	X
IsHitTestVisible	X
IsManipulationEnabled	X
IsSharedSizeScope	X
IsTabStop	X
Localization	X
Margin	X
MaxHeight	X
MaxWidth	X
MinHeight	X
MinWidth	X
Name	X
Opacity	X
OpacityMask	X

Property	Animated Fire
OverridesDefaultStyle	X
Padding	X
RenderTransformOrigin	X
Row	X
RowSpan	X
SnapsToDevicePixels	X
Style	X
TabIndex	X
Tag	X
Template	X
ToolTip	X
Uid	X
UseLayoutRounding	X
VerticalAlignment	X
VerticalContentAlignment	X
Visibility	X
Width	X

Miscellaneous Controls Property Descriptions

The following table shows the properties descriptions used by all control:

Property	Description
AllowDrop	Select this check box to indicate that the control can accept data that is dragged onto it.
Background	Expand this field to select the background color of the selected control.
BorderBrush	Expand this field to select the border color of the selected control.

Property	Description
BorderThickness	Specifies the border thickness of a control, in pixels.
ClipToBounds	Select this check box to clip the content of this control (or content coming from the child elements of this control) to fit into the size of the containing control.
Column	Specifies the column within the grid where the control is located.
ColumnSpan	Specifies the total number of columns that child content spans within a grid.
Effect	<p>Click the arrow to expand the property, select the effect you want to apply to the control, and then configure the specific values for the effect. Valid values to select are:</p> <ul style="list-style-type: none"> • Blur <ul style="list-style-type: none"> ◦ Kernel Type ◦ Radius • Drop Shadow <ul style="list-style-type: none"> ◦ Color ◦ Direction ◦ Opacity ◦ Shadow Depth
FlameIsOn	Select this check box to indicate that the flame is on.
FlameMaxHeight	Specifies the maximum height of the animated flame.
FlameMinHeight	Specifies the minimum height of the animated flame.
FlickerFadeSpeed	Specifies the speed at which the flame fades.
FlickerSpeed	Specifies the speed at which the flame flickers.

Property	Description
FlowDirection	<p>Specifies the direction that text and other user interface (UI) elements flow within any parent control. Selections are:</p> <ul style="list-style-type: none"> • Left to Right • Right to Left
Focusable	<p>Select this check box to indicate that the control can receive focus.</p>
FocusVisualStyle	<p>Specifies a property that enables customization of appearance, effects, or other style characteristics that apply to this control when it has focus.</p>
FontFamily	<p>Specifies the font family for the control.</p>
FontSize	<p>Specifies the font size for the control.</p>
FontStretch	<p>Specifies the degree to which a font is condensed or expanded on the screen.</p>
FontStyle	<p>Specifies the font style for the control.</p>
FontWeight	<p>Specifies the weight or thickness of the specified font.</p>
ForceCursor	<p>Select this check box to indicate that the user interface (UI) renders the cursor as declared by the Cursor property.</p>
Foreground	<p>Expand this field to select the foreground color.</p>
Height	<p>Specifies the height of the control, in pixels.</p>
HorizontalAlignment	<p>Specifies the horizontal alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch

Property	Description
HorizontalContentAlignment	<p>Specifies the horizontal alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
ImageStretch	<p>Specifies how the image should be stretched to fill the destination rectangle. Selections are:</p> <ul style="list-style-type: none"> • Fill • None • Uniform • UniformToFill
IsEnabled	<p>Select this check box to indicate that this control is enabled in the user interface (UI).</p>
IsHitTestVisible	<p>Select this check box to indicate that the control can possibly be returned as a hit test result from some portion of its rendered content.</p>
IsManipulationEnabled	<p>Select this check box to indicate that the manipulation events for the control are enabled.</p>
IsSharedSizeScope	<p>Select this check box to indicate that controls are sharing size information.</p>
IsTabStop	<p>Select this check box to indicate that a control is included in tab navigation.</p>
Localization	<p>Click the ellipsis button to open the Localization Configuration dialog box where you can configure the property to localize, the language to use for localization, and the text to localize.</p>
Margin	<p>Specifies the outer margin of a control.</p>
MaxHeight	<p>Specifies the maximum height of a control.</p>
MaxWidth	<p>Specifies the maximum width of a control.</p>

Property	Description
MinHeight	Specifies the minimum height of a control.
MinWidth	Specifies the minimum width of a control.
Name	Specifies the identifying name of the control. The name provides a reference so that code behind, such as event handler code, can refer to a markup element during processing by a XAML processor.
Opacity	Specifies the opacity factor applied to the entire control. Valid values are 0 through 1, with up to 15 decimal places.
OpacityMask	Expand this field to select a color for the opacity mask, which is applied to any alpha-channel masking for the rendered content of this control.
OverridesDefaultStyle	Select this check box to indicate that this control incorporates style properties from theme styles.
Padding	Specifies the amount of padding to apply to the inside edge of each side of a control.
RenderTransformOrigin	<p>Specifies the center point of any possible render transform, relative to the bounds of the element.</p> <ul style="list-style-type: none"> • X is the horizontal origin factor, given as a value between 0 and 1. • Y is the vertical origin factor, given as a value between 0 and 1.
Row	RowBackground
RowSpan	Specifies the total number of rows that child content spans within a grid.
SnapsToDevicePixels	Select this check box to indicate that rendering for this control uses device-specific pixel settings.
Style	Specifies the style used by this control when it is rendered.

Property	Description
TabIndex	Specifies the order in which controls receive focus when the Tab key is pressed.
Tag	Specifies an arbitrary object value that can be used to store custom information about this control.
Template	Click the ellipsis button to open the Open dialog box to select a template to use.
ToolTip	Specifies the tool-tip that displays for this control in the user interface (UI).
Uid	Specifies the unique identifier (for localization) for this control.
UseLayoutRounding	Specifies a value that determines whether rendering for the object and its visual subtree should use rounding behavior that aligns rendering to whole pixels.
VerticalAlignment	<p>Specifies the vertical alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
VerticalContentAlignment	<p>Specifies the vertical alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
Visibility	Specifies the visibility of this control. Selections are:

Property	Description
	<ul style="list-style-type: none"> • Collapsed • Hidden • Visible
Width	Specifies the width of the control, in pixels.

Animated Fire



The Animated Fire control simulates fire in an industrial setting.

See **Miscellaneous Controls** to view the usage and description of the properties for this control.

Related reference

[Miscellaneous Controls \(on page 990\)](#)

Motor and Fan Controls

Motors controls are graphical representations of motors, fans, and pumps, and rotary controls.

Motors provide a graphic illustration of how fast a motor is running by showing an indicator brush in the control. The speed of the motor is user-controlled; the faster the indicator moves, the faster the motor is running. By manipulating the properties, motor controls can be used to monitor motor behavior, lifespan, or even scheduled maintenance intervals.

Fan controls display the relative speed of rotation of the fan blades. Fans can be used as an independent control, or be added to a motor to simulate a motor fan.

When a motor or fan control receives a null value for the IsOn property, the control sets its Opacity property to 0.2. The control appears disabled until a True or False value is received for the IsOn property.

Rotary controls contain circular graphics that spin 360 degrees at user-defined rotation speeds. When a rotary control receives a null value for the IsOn property, the control sets its Opacity property to 0.2. The control appears disabled until a True or False value is received for the IsOn property.

Motor and Fan Controls Property Usage

The following table shows the properties used by each control:

Prop- erty	Cen- trifu- gal Pump	Gear	Heavy Motor	Heli- cal Motor	Metal Fly Wheel	Multi Blade Fan	Pad- dle Wheel	Sim- ple Fan	Small Fly Wheel	Sprock- et	Square Motor	Tire	Wind Gen- era- tor
Uid	X	X	X	X	X	X	X	X	X	X	X	X	X
Use- Lay- out- Round- ing	X	X	X	X	X	X	X	X	X	X	X	X	X
Ver- tical- Align- ment	X	X	X	X	X	X	X	X	X	X	X	X	X
Ver- tical- Con- tent- Align- ment	X		X	X	X	X	X	X	X	X	X	X	X
Visi- bility	X	X	X	X	X	X	X	X	X	X	X	X	X
Width	X	X	X	X	X	X	X	X	X	X	X	X	X

Motor and Fan Controls Property Descriptions

The following table shows the properties descriptions used by all control:

Property	Description
AllowDrop	Select this check box to indicate that the control can accept data that is dragged onto it.
Background	Expand this field to select the background color of the selected control.
BladeBrush	Expand this field to select the color of the blade.

Property	Description
BladeStrokeThickness	Specifies the thickness of the blade.
BorderBrush	Expand this field to select the border color of the selected control.
BorderThickness	Specifies the border thickness of a control, in pixels.
CenterCapBrush	Expand this field to select a color for the center cap.
ClipToBounds	Select this check box to clip the content of this control (or content coming from the child elements of this control) to fit into the size of the containing control.
Column	Specifies the column within the grid where the control is located.
ColumnSpan	Specifies the total number of columns that child content spans within a grid.
Effect	<p>Click the arrow to expand the property, select the effect you want to apply to the control, and then configure the specific values for the effect. Valid values to select are:</p> <ul style="list-style-type: none"> • Blur <ul style="list-style-type: none"> ◦ Kernel Type ◦ Radius • Drop Shadow <ul style="list-style-type: none"> ◦ Color ◦ Direction ◦ Opacity ◦ Shadow Depth
FlowDirection	Specifies the direction that text and other user interface (UI) elements flow within any parent control. Selections are:

Property	Description
	<ul style="list-style-type: none"> • Left to Right • Right to Left
Focusable	Select this check box to indicate that the control can receive focus.
FocusVisualStyle	Specifies a property that enables customization of appearance, effects, or other style characteristics that apply to this control when it has focus.
FontFamily	Specifies the font family for the control.
FontSize	Specifies the font size for the control.
FontStretch	Specifies the degree to which a font is condensed or expanded on the screen.
FontStyle	Specifies the font style for the control.
FontWeight	Specifies the weight or thickness of the specified font.
ForceCursor	Select this check box to indicate that the user interface (UI) renders the cursor as declared by the Cursor property.
Foreground	Expand this field to select the foreground color.
GlareBrush	Expand this field to specify the glare effect for the control.
Height	Specifies the height of the control, in pixels.
HorizontalAlignment	<p>Specifies the horizontal alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
HorizontalContentAlignment	Specifies the horizontal alignment of the control's content. Selections are:

Property	Description
	<ul style="list-style-type: none"> • Left • Center • Right • Stretch
ImageStretch	<p>Specifies how the image should be stretched to fill the destination rectangle. Selections are:</p> <ul style="list-style-type: none"> • Fill • None • Uniform • UniformToFill
IndicatorBrush	Expand this field to select the color to use for the graphical indicator (pulse) of the motor speed.
InnerBorderBrush	Expand this field to select the color for the inner border.
IsEnabled	Select this check box to indicate that this control is enabled in the user interface (UI).
IsHitTestVisible	Select this check box to indicate that the control can possibly be returned as a hit test result from some portion of its rendered content.
IsManipulationEnabled	Select this check box to indicate that the manipulation events for the control are enabled.
IsOn	Select this check box to indicate that the control is turned on.
IsSharedSizeScope	Select this check box to indicate that controls are sharing size information.
IsTabStop	Select this check box to indicate that a control is included in tab navigation.
Localization	Click the ellipsis button to open the Localization Configuration dialog box where you can configure the property to localize, the language to use for localization, and the text to localize.

Property	Description
Margin	Specifies the outer margin of a control.
MaxHeight	Specifies the maximum height of a control.
MaxWidth	Specifies the maximum width of a control.
MinHeight	Specifies the minimum height of a control.
MinorBackground	Expand this field to select the fill color of secondary components on the control.
MinWidth	Specifies the minimum width of a control.
MotorSpeed	Specifies the relative motor speed of the control.
Name	Specifies the identifying name of the control. The name provides a reference so that code behind, such as event handler code, can refer to a markup element during processing by a XAML processor.
Opacity	Specifies the opacity factor applied to the entire control. Valid values are 0 through 1, with up to 15 decimal places.
OpacityMask	Expand this field to select a color for the opacity mask, which is applied to any alpha-channel masking for the rendered content of this control.
OverridesDefaultStyle	Select this check box to indicate that this control incorporates style properties from theme styles.
Padding	Specifies the amount of padding to apply to the inside edge of each side of a control.
RenderTransformOrigin	<p>Specifies the center point of any possible render transform, relative to the bounds of the element.</p> <ul style="list-style-type: none"> • X is the horizontal origin factor, given as a value between 0 and 1. • Y is the vertical origin factor, given as a value between 0 and 1.
RotationSpeed	Specifies the rotation speed of the control.

Property	Description
Row	RowBackground
RowSpan	Specifies the total number of rows that child content spans within a grid.
ShowGridLines	Select this check box to indicate that grid lines are visible within this grid.
SnapsToDevicePixels	Select this check box to indicate that rendering for this control uses device-specific pixel settings.
StrokeThickness	Specifies the width of the control's outline.
Style	Specifies the style used by this control when it is rendered.
TabIndex	Specifies the order in which controls receive focus when the Tab key is pressed.
Tag	Specifies an arbitrary object value that can be used to store custom information about this control.
Template	Click the ellipsis button to open the Open dialog box to select a template to use.
TertiaryBackground	Expand this field to select the inner background color.
TertiaryBorderBrush	Expand this field to select the innermost border color of the selected control.
TertiaryStrokeThickness	Specifies the width of the control's innermost outline.
ToolTip	Specifies the tool-tip that displays for this control in the user interface (UI).
Uid	Specifies the unique identifier (for localization) for this control.
UseLayoutRounding	Specifies a value that determines whether rendering for the object and its visual subtree should use rounding behavior that aligns rendering to whole pixels.

Property	Description
VerticalAlignment	Specifies the vertical alignment characteristics applied to this control. Selections are: <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
VerticalContentAlignment	Specifies the vertical alignment of the control's content. Selections are: <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
Visibility	Specifies the visibility of this control. Selections are: <ul style="list-style-type: none"> • Collapsed • Hidden • Visible
Width	Specifies the width of the control, in pixels.

Centrifugal Pump



The Centrifugal Pump control is a roto-dynamic control with a rotating inner region simulating the action of an impeller.

The rotational speed is determined by the value assigned to the MotorSpeed property.

See **Motor and Fan Controls** to view the usage and description of the properties for this control.

Related reference

[Motor and Fan Controls \(on page 998\)](#)

Gear



The Gear control is a profiled wheel with meshable teeth, featuring adjustable rotation speeds and direction.

See **Motor and Fan Controls** to view the usage and description of the properties for this control.

Related reference

[Motor and Fan Controls \(on page 998\)](#)

Heavy Motor



The Heavy Motor control has a flashing inner region that simulates the rotation of a motor. This control has the appearance of a large industrial motor.

See **Motor and Fan Controls** to view the usage and description of the properties for this control.

Related reference

[Motor and Fan Controls \(on page 998\)](#)

Helical Motor



The Helical Motor control has a flashing inner region that simulates the rotation of a motor. The cylindrical body shape is identified with a helical motor.

See **Motor and Fan Controls** to view the usage and description of the properties for this control.

Related reference

[Motor and Fan Controls \(on page 998\)](#)

Metal Fly Wheel



The Metal Fly Wheel control is a profiled wheel with eight counter-balance holes, featuring adjustable rotation speeds and direction.

See **Motor and Fan Controls** to view the usage and description of the properties for this control.

Related reference

[Motor and Fan Controls \(on page 998\)](#)

Multi Blade Fan



The Multi Blade Fan control has eight rotating fan blades in an enclosure. The rotational speed is determined by the control's Value property.

See **Motor and Fan Controls** to view the usage and description of the properties for this control.

Related reference

[Motor and Fan Controls \(on page 998\)](#)

Paddle Wheel



The Paddle Wheel control is a wheel with twelve paddles, featuring adjustable rotation speeds and direction, typically used for mixing.

See **Motor and Fan Controls** to view the usage and description of the properties for this control.

Related reference

[Motor and Fan Controls \(on page 998\)](#)

Simple Fan



The Simple Fan control has four rotating fan blades. The rotational speed is determined by the control's Value property.

See **Motor and Fan Controls** to view the usage and description of the properties for this control.

Related reference

[Motor and Fan Controls \(on page 998\)](#)

Small Fly Wheel



The Small Fly Wheel control is an enclosed profiled wheel, similar to the Metal Fly Wheel control, featuring adjustable rotation speeds and direction.

See **Motor and Fan Controls** to view the usage and description of the properties for this control.

Related reference

[Motor and Fan Controls \(on page 998\)](#)

Sprocket



The Sprocket control is a profiled wheel with blade-style non-meshable teeth, featuring adjustable rotation speeds and direction.

See **Motor and Fan Controls** to view the usage and description of the properties for this control.

Related reference

[Motor and Fan Controls \(on page 998\)](#)

Square Motor



The Square Motor control has a flashing inner region that simulates the rotation speed of a motor.

See **Motor and Fan Controls** to view the usage and description of the properties for this control.

Related reference

[Motor and Fan Controls \(on page 998\)](#)

Tire



The Tire control is a combination of an outer tire, rim, and hubcap, featuring adjustable rotation speeds and direction.

See **Motor and Fan Controls** to view the usage and description of the properties for this control.

Related reference

[Motor and Fan Controls \(on page 998\)](#)

Wind Generator



The Wind Generator control represents a large wind power-generator. The rotational speed of the three fan blades is determined by the control's Value property.

See **Motor and Fan Controls** to view the usage and description of the properties for this control.

Related reference

[Motor and Fan Controls \(on page 998\)](#)

Pipe and Tank Controls

Pipe controls are used to display the directional movement of vapors, liquids, or semi-solids. Tank controls display the level of the tank content relative to the top or bottom, and as a percent of the total tank capacity level.

Pipe controls contain a flow indicator to graphically display the direction the vapor or liquid is moving.

When a tank control receives a value outside the specified range (that is, above the Maximum or below the Minimum), or the value is Null, the control sets its Opacity property to 0.2. The control appears disabled until the value is back within the specified range or is no longer Null.



Note:

In the Silverlight forms designer, the pipe and tank controls are grouped separately.

Pipe and Tank Controls Property Usage

The following table shows the properties used by each control:

Property	Connector Pipe	Cut-away Tank	Cylinder Tank	Fitted Elbow Pipe	Fitted Pipe	Fitted Pipe Marquee	Funnel Tank	Indicator Tank	Ladder Tank	Mixing Tank	Standard Elbow Pipe	Standard Pipe	Standard Pipe Marquee	Universal Pipe Connector	Valve	Vessel Tank
Allow-Drop	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Animation-Speed						X							X			

Property	Con- nec- tor Pipe	Cut- away Tank	Cylin- der Tank	Fit- ted El- bow Pipe	Fit- ted Pipe	Fit- ted Pipe Mar- quee	Fun- nel Tank	Indi- ca- tor Tank	Lad- der Tank	Mix- ing Tank	Stan- dard El- bow Pipe	Stan- dard Pipe	Stan- dard Pipe Mar- quee	Uni- ver- sal Pipe Con- nec- tor	Valve	Ves- sel Tank
Use- Lay- out- Round- ing	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Val- ue			X		X		X	X	X	X						
Ver- ti- cal- Align- ment	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Ver- ti- cal- Con- tent- Align- ment	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Visi- bility	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Width	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Pipe and Tank Controls Property Descriptions

The following table shows the properties descriptions used by all control:

Property	Description
AllowDrop	Select this check box to indicate that the control can accept data that is dragged onto it.
Background	Expand this field to select the background color of the selected control.
BorderBrush	Expand this field to select the border color of the selected control.
BorderThickness	Specifies the border thickness of a control, in pixels.
ClipToBounds	Select this check box to clip the content of this control (or content coming from the child elements of this control) to fit into the size of the containing control.
CloseBrush	Expand this field to select a color to indicate that the valve is closed
Column	Specifies the column within the grid where the control is located.
ColumnSpan	Specifies the total number of columns that child content spans within a grid.
CornerRadius	Specifies the degree to which the corners of a control are rounded.
Effect	<p>Click the arrow to expand the property, select the effect you want to apply to the control, and then configure the specific values for the effect. Valid values to select are:</p> <ul style="list-style-type: none"> • Blur <ul style="list-style-type: none"> ◦ Kernel Type ◦ Radius • Drop Shadow <ul style="list-style-type: none"> ◦ Color ◦ Direction

Property	Description
	<ul style="list-style-type: none"> ◦ Opacity ◦ Shadow Depth
FlowDirection	<p>Specifies the direction that text and other user interface (UI) elements flow within any parent control. Selections are:</p> <ul style="list-style-type: none"> • Left to Right • Right to Left
FluidBrush	Expand this field to select the color for the fluid portion of the selected control.
Focusable	Select this check box to indicate that the control can receive focus.
FocusVisualStyle	Specifies a property that enables customization of appearance, effects, or other style characteristics that apply to this control when it has focus.
FontFamily	Specifies the font family for the control.
FontSize	Specifies the font size for the control.
FontStretch	Specifies the degree to which a font is condensed or expanded on the screen.
FontStyle	Specifies the font style for the control.
FontWeight	Specifies the weight or thickness of the specified font.
ForceCursor	Select this check box to indicate that the user interface (UI) renders the cursor as declared by the Cursor property.
Foreground	Expand this field to select the foreground color.
GlareBrush	Expand this field to specify the glare effect for the control.
Height	Specifies the height of the control, in pixels.
HorizontalAlignment	Specifies the horizontal alignment characteristics applied to this control. Selections are:

Property	Description
	<ul style="list-style-type: none"> • Left • Center • Right • Stretch
HorizontalContentAlignment	<p>Specifies the horizontal alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
ImageStretch	<p>Specifies how the image should be stretched to fill the destination rectangle. Selections are:</p> <ul style="list-style-type: none"> • Fill • None • Uniform • UniformToFill
IsEnabled	<p>Select this check box to indicate that this control is enabled in the user interface (UI).</p>
IsHitTestVisible	<p>Select this check box to indicate that the control can possibly be returned as a hit test result from some portion of its rendered content.</p>
IsManipulationEnabled	<p>Select this check box to indicate that the manipulation events for the control are enabled.</p>
IsOn	<p>Select this check box to indicate that the control is turned on.</p>
IsSharedSizeScope	<p>Select this check box to indicate that controls are sharing size information.</p>
IsTabStop	<p>Select this check box to indicate that a control is included in tab navigation.</p>
Localization	<p>Click the ellipsis button to open the Localization Configuration dialog box where you can configure</p>

Property	Description
	the property to localize, the language to use for localization, and the text to localize.
Margin	Specifies the outer margin of a control.
MarqueePauseOnMouseHover	Select this check box to indicate that the marquee pauses when the mouse hovers over it.
MaxHeight	Specifies the maximum height of a control.
Maximum	Specifies the maximum value of the range of the control.
MaxWidth	Specifies the maximum width of a control.
MinHeight	Specifies the minimum height of a control.
Minimum	Specifies the minimum value of the range of the control.
MinorBackground	Expand this field to select the fill color of secondary components on the control.
MinWidth	Specifies the minimum width of a control.
MixingSpeed	Specifies the rotation speed of the mixing paddle.
Name	Specifies the identifying name of the control. The name provides a reference so that code behind, such as event handler code, can refer to a markup element during processing by a XAML processor.
Opacity	Specifies the opacity factor applied to the entire control. Valid values are 0 through 1, with up to 15 decimal places.
OpacityMask	Expand this field to select a color for the opacity mask, which is applied to any alpha-channel masking for the rendered content of this control.
OpenBrush	Expand this field to select a color to indicate that the valve is open.
OverridesDefaultStyle	Select this check box to indicate that this control incorporates style properties from theme styles.

Property	Description
Padding	Specifies the amount of padding to apply to the inside edge of each side of a control.
RenderTransformOrigin	<p>Specifies the center point of any possible render transform, relative to the bounds of the element.</p> <ul style="list-style-type: none"> • X is the horizontal origin factor, given as a value between 0 and 1. • Y is the vertical origin factor, given as a value between 0 and 1.
Row	RowBackground
RowSpan	Specifies the total number of rows that child content spans within a grid.
SnapsToDevicePixels	Select this check box to indicate that rendering for this control uses device-specific pixel settings.
StrokeThickness	Specifies the width of the control's outline.
Style	Specifies the style used by this control when it is rendered.
TabIndex	Specifies the order in which controls receive focus when the Tab key is pressed.
Tag	Specifies an arbitrary object value that can be used to store custom information about this control.
TankShape	Specifies the shape of the tank.
Template	Click the ellipsis button to open the Open dialog box to select a template to use.
TertiaryBackground	Expand this field to select the inner background color.
ToolTip	Specifies the tool-tip that displays for this control in the user interface (UI).
Uid	Specifies the unique identifier (for localization) for this control.

Property	Description
UseLayoutRounding	Specifies a value that determines whether rendering for the object and its visual subtree should use rounding behavior that aligns rendering to whole pixels.
Value	Specifies the current magnitude of the range control.
VerticalAlignment	<p>Specifies the vertical alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
VerticalContentAlignment	<p>Specifies the vertical alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
Visibility	<p>Specifies the visibility of this control. Selections are:</p> <ul style="list-style-type: none"> • Collapsed • Hidden • Visible
Width	Specifies the width of the control, in pixels.

Connector Pipe



The Connector Pipe control is a fitted tee connector that allows pipe directions to change or merge in a three-way 90 degree T-shaped connection.

See **Pipe and Tank Controls** to view the usage and description of the properties for this control.

Related reference

[Pipe and Tank Controls \(on page 1015\)](#)

Cutaway Tank



The Cutaway Tank control is an enclosed vessel featuring a cutaway view and an adjustable fill level.

See **Pipe and Tank Controls** to view the usage and description of the properties for this control.

Related reference

[Pipe and Tank Controls \(on page 1015\)](#)

Cylinder Tank



The Cylinder Tank control is a simple holding vessel with an adjustable fill level.

See **Pipe and Tank Controls** to view the usage and description of the properties for this control.

Related reference

[Pipe and Tank Controls \(on page 1015\)](#)

Fitted Elbow Pipe



The Fitted Elbow Pipe control is a length of standard pipe set at a 90 degree angle, with the addition of larger cuffs at either end, changing the direction of the pipe.

See **Pipe and Tank Controls** to view the usage and description of the properties for this control.

Related reference

[Pipe and Tank Controls \(on page 1015\)](#)

Fitted Pipe



The Fitted Pipe control is a length of standard pipe with the addition of larger cuffs at either end.

See **Pipe and Tank Controls** to view the usage and description of the properties for this control.

Related reference

[Pipe and Tank Controls \(on page 1015\)](#)

Fitted Pipe Marquee



The Fitted Pipe Marquee control is a fitted pipe with a marquee control in the center. This pipe can be used to show the flow direction or contents of the pipe by using scrolling content.

See **Pipe and Tank Controls** to view the usage and description of the properties for this control.

Related reference

[Pipe and Tank Controls \(on page 1015\)](#)

Funnel Tank



The Funnel Tank control is a simple vessel that can channel content from a larger stream into a graduated smaller stream.

See **Pipe and Tank Controls** to view the usage and description of the properties for this control.

Related reference

[Pipe and Tank Controls \(on page 1015\)](#)

Indicator Tank



The Indicator Tank control is a holding vessel that features a see-through panel to view the fluid level, and an adjustable fill level.

See **Pipe and Tank Controls** to view the usage and description of the properties for this control.

Related reference

[Pipe and Tank Controls \(on page 1015\)](#)

Ladder Tank



The Ladder Tank control is a storage vessel inside an enclosure with an adjustable fill level.

See **Pipe and Tank Controls** to view the usage and description of the properties for this control.

Related reference

[Pipe and Tank Controls \(on page 1015\)](#)

Mixing Tank



The Mixing Tank control is a tank with a mixing paddle that rotates clockwise. A mixing tank has an adjustable motor speed and an adjustable fill level.

See **Pipe and Tank Controls** to view the usage and description of the properties for this control.

Related reference

[Pipe and Tank Controls \(on page 1015\)](#)

Standard Elbow Pipe



The Standard Elbow Pipe control is a length of standard pipe set at a 90 degree angle, changing the direction of the pipe.

See **Pipe and Tank Controls** to view the usage and description of the properties for this control.

Related reference

[Pipe and Tank Controls \(on page 1015\)](#)

Standard Pipe



The Standard Pipe control is a length of cylindrical tubing used to direct substances from one area to another.

See **Pipe and Tank Controls** to view the usage and description of the properties for this control.

Related reference

[Pipe and Tank Controls \(on page 1015\)](#)

Standard Pipe Marquee



The Standard Pipe Marquee control is a length of standard pipe with a marquee control in the center. This control can be used to show the flow direction and contents of the pipe with scrolling content.

See **Pipe and Tank Controls** to view the usage and description of the properties for this control.

Related reference

[Pipe and Tank Controls \(on page 1015\)](#)

Universal Pipe Connector



The Universal Pipe Connector is a spherical disk that can connect varying amounts of pipes at varying degrees to create specific configurations.

See **Pipe and Tank Controls** to view the usage and description of the properties for this control.

Related reference

[Pipe and Tank Controls \(on page 1015\)](#)

Valve



The Valve control is used to control the open or closed status of a pipe. Its orientation can be changed so that it can be used with a pipe that is in a horizontal or vertical position.

See **Pipe and Tank Controls** to view the usage and description of the properties for this control.

Related reference

[Pipe and Tank Controls \(on page 1015\)](#)

Vessel Tank



The Vessel Tank control is an enclosed, horizontally oriented holding vessel with an adjustable fill level.

See **Pipe and Tank Controls** to view the usage and description of the properties for this control.

Related reference

[Pipe and Tank Controls \(on page 1015\)](#)

Custom Controls

The **Designer** provides the following group of custom controls.

Custom Controls Property Usage

The following table shows the properties used by each control:

Property	Universal Browser	User Validation
AllowDrop	X	X
Authenticated		X
Background	X	X
BorderBrush	X	X
BorderThickness	X	X
ClipToBounds	X	X
Column	X	X
ColumnSpan	X	X
DataFilter	X	
Effect	X	X
FlowDirection	X	X
Focusable	X	X
FocusVisualStyle	X	X
FontFamily	X	X
FontSize	X	X
FontStretch	X	X
FontStyle	X	X
FontWeight	X	X
ForceCursor	X	X
Foreground	X	X
Height	X	X
HorizontalAlignment	X	X
HorizontalContentAlignment	X	X
InstructionText	X	

Property	Universal Browser	User Validation
IsEnabled	X	X
IsHitTestVisible	X	X
IsManipulationEnabled	X	X
IsSharedSizeScope	X	X
IsTabStop	X	X
ItemText	X	
Localization	X	X
Margin	X	X
MaxHeight	X	X
MaxWidth	X	X
MinHeight	X	X
MinWidth	X	X
ModelsAndEntryPointFilter	X	
MultipleSelectAllowed	X	
Name	X	X
Opacity	X	X
OpacityMask	X	X
OverridesDefaultStyle	X	X
Padding	X	X
RenderTransformOrigin	X	X
Row	X	X
RowSpan	X	X
SnapsToDevicePixels	X	X
StrokeThickness	X	
Style	X	X
TabIndex	X	X

Property	Universal Browser	User Validation
Tag	X	X
Template	X	X
ToolTip	X	X
Uid	X	X
UseLayoutRounding	X	X
VerticalAlignment	X	X
VerticalContentAlignment	X	X
Visibility	X	X
Width	X	X

Custom Controls Property Descriptions

The following table shows the properties descriptions used by all control:

Property	Description
AllowDrop	Select this check box to indicate that the control can accept data that is dragged onto it.
Authenticated	This check box is selected when a user's authentication attempt succeeds.
Background	Expand this field to select the background color of the selected control.
BorderBrush	Expand this field to select the border color of the selected control.
BorderThickness	Specifies the border thickness of a control, in pixels.
ClipToBounds	Select this check box to clip the content of this control (or content coming from the child elements of this control) to fit into the size of the containing control.
Column	Specifies the column within the grid where the control is located.

Property	Description
ColumnSpan	Specifies the total number of columns that child content spans within a grid.
DataFilter	Click the ellipsis button to open the Browser Data Filter dialog box where you can filter the data to display in the Universal Browser control. The settings you configure in this dialog box are ignored if the ShowModels property check box is selected.
Effect	<p>Click the arrow to expand the property, select the effect you want to apply to the control, and then configure the specific values for the effect. Valid values to select are:</p> <ul style="list-style-type: none"> • Blur <ul style="list-style-type: none"> ◦ Kernel Type ◦ Radius • Drop Shadow <ul style="list-style-type: none"> ◦ Color ◦ Direction ◦ Opactiy ◦ Shadow Depth
FlowDirection	<p>Specifies the direction that text and other user interface (UI) elements flow within any parent control. Selections are:</p> <ul style="list-style-type: none"> • Left to Right • Right to Left
Focusable	Select this check box to indicate that the control can receive focus.
FocusVisualStyle	Specifies a property that enables customization of appearance, effects, or other style characteristics that apply to this control when it has focus.
FontFamily	Specifies the font family for the control.
FontSize	Specifies the font size for the control.

Property	Description
FontStretch	Specifies the degree to which a font is condensed or expanded on the screen.
FontStyle	Specifies the font style for the control.
FontWeight	Specifies the weight or thickness of the specified font.
ForceCursor	Select this check box to indicate that the user interface (UI) renders the cursor as declared by the Cursor property.
Foreground	Expand this field to select the foreground color.
GlareBrush	Expand this field to specify the glare effect for the control.
Height	Specifies the height of the control, in pixels.
HorizontalAlignment	<p>Specifies the horizontal alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
HorizontalContentAlignment	<p>Specifies the horizontal alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
ImageStretch	<p>Specifies how the image should be stretched to fill the destination rectangle. Selections are:</p> <ul style="list-style-type: none"> • Fill • None

Property	Description
	<ul style="list-style-type: none"> • Uniform • UniformToFill
InstructionText	Specifies the item description displayed initially when nothing is selected. The default text is "Please select an appropriate item."
IsEnabled	Select this check box to indicate that this control is enabled in the user interface (UI).
IsHitTestVisible	Select this check box to indicate that the control can possibly be returned as a hit test result from some portion of its rendered content.
IsManipulationEnabled	Select this check box to indicate that the manipulation events for the control are enabled.
IsSharedSizeScope	Select this check box to indicate that controls are sharing size information.
IsTabStop	Select this check box to indicate that a control is included in tab navigation.
ItemText	Specifies the item name displayed initially when nothing is selected. The default text is "Not Set."
Localization	Click the ellipsis button to open the Localization Configuration dialog box where you can configure the property to localize, the language to use for localization, and the text to localize.
Margin	Specifies the outer margin of a control.
MaxHeight	Specifies the maximum height of a control.
MaxWidth	Specifies the maximum width of a control.
MinHeight	Specifies the minimum height of a control.
MinWidth	Specifies the minimum width of a control.
ModelsAndEntryPointFilter	Click the ellipsis button to open the Browser Models and Entry Point Filter dialog box where you can specify a list of the models and entry points to be

Property	Description
	<p>included in, or excluded from, the Universal Browser when it appears. By default, all models and entry points are displayed. In order for the selected models and entry points to display, the ShowModels property check box must be selected.</p> <div data-bbox="824 512 1422 779" style="border: 1px solid #00a0c0; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note: When you exclude entry points, you must also exclude the "All" entry point, if it exists. Including the "All" entry point takes precedence over individual exclusions.</p> </div>
MultipleSelectAllowed	Select this check box to allow multiple resources to be selected in the Universal Browser.
Name	Specifies the identifying name of the control. The name provides a reference so that code behind, such as event handler code, can refer to a markup element during processing by a XAML processor.
Opacity	Specifies the opacity factor applied to the entire control. Valid values are 0 through 1, with up to 15 decimal places.
OpacityMask	Expand this field to select a color for the opacity mask, which is applied to any alpha-channel masking for the rendered content of this control.
OverridesDefaultStyle	Select this check box to indicate that this control incorporates style properties from theme styles.
Padding	Specifies the amount of padding to apply to the inside edge of each side of a control.
RenderTransformOrigin	Specifies the center point of any possible render transform, relative to the bounds of the element.

Property	Description
	<ul style="list-style-type: none"> • X is the horizontal origin factor, given as a value between 0 and 1. • Y is the vertical origin factor, given as a value between 0 and 1.
Row	RowBackground
RowSpan	Specifies the total number of rows that child content spans within a grid.
SnapsToDevicePixels	Select this check box to indicate that rendering for this control uses device-specific pixel settings.
StrokeThickness	Specifies the width of the control's outline.
Style	Specifies the style used by this control when it is rendered.
TabIndex	Specifies the order in which controls receive focus when the Tab key is pressed.
Tag	Specifies an arbitrary object value that can be used to store custom information about this control.
Template	Click the ellipsis button to open the Open dialog box to select a template to use.
ToolTip	Specifies the tool-tip that displays for this control in the user interface (UI).
Uid	Specifies the unique identifier (for localization) for this control.
UseLayoutRounding	Specifies a value that determines whether rendering for the object and its visual subtree should use rounding behavior that aligns rendering to whole pixels.
VerticalAlignment	<p>Specifies the vertical alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center

Property	Description
	<ul style="list-style-type: none"> • Top • Stretch
VerticalContentAlignment	<p>Specifies the vertical alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
Visibility	<p>Specifies the visibility of this control. Selections are:</p> <ul style="list-style-type: none"> • Collapsed • Hidden • Visible
Width	Specifies the width of the control, in pixels.

Universal Browser



The Universal Browser is a WPF control that allows you to add the Universal Browser to a form or user display.

The Show Models and ModelsAndEntryPointFilter properties work together to allow you to select the models and/or entry points to display in the Universal Browser. If the ShowModels check box is cleared, your selections in the ModelsAndEntryPointsFilter dialog box are ignored. In that case, you can use the DataFilter property to set the models.

The Universal Browser control functions in the same manner as the Universal Browser when it is opened in the Workflow client; that is, hierarchical models expand as you select resources, basic and advanced searches can be performed, and Quick Find is available. For more information, see Universal Browser.

See **Custom Controls** to view the usage and description of the properties for this control.

Related reference

[Custom Controls \(on page 1033\)](#)

User Validation



The User Validation is a WPF control that allows you to specify a username and password, and validates that it is a Workflow user. It is made up of a Validating Text Box control and a Password Box control.

If you use Windows user names, the User ID text box must be prefaced with your domain; for example, <domain name>\username. The red border around the User Id text box remains until the user name is authenticated.

See **Custom Controls** to view the usage and description of the properties for this control.

Related reference

[Custom Controls \(on page 1033\)](#)

Scrolling Controls

Scrolling controls provide the ability to display text in a scrolling manner.

Scrolling Controls Property Usage

The following table shows the properties used by each control:

Property	Area Navigator	Fade Scroller	Marquee	News Scroller
AllowDrop	X	X	X	X
AlternationCount	X			
Background	X	X	X	X
BorderBrush	X	X	X	X
BorderThickness	X	X	X	X
CacheMode		X	X	X
Clip		X	X	X
ClipToBounds	X			
Column	X	X	X	X
ColumnSpan	X	X	X	X
CornerRadius		X	X	X

Property	Area Navigator	Fade Scroller	Marquee	News Scroller
Cursor		X	X	X
DataContext		X	X	X
DisplayMember-Path	X	X	X	
Effect	X	X	X	X
Enable Rotation	X			
FadeScrollerAnimationSpeed		X		
FadeScrollerAnimationType		X		
FadeScrollerDestinationType		X		
FadeScrollerFadeInSpeed		X		
FadeScrollerFadeOutSpeed		X		
FadeScrollerPauseOnMouseHover		X		
FeedRefreshInterval				X
FeedUri				X
FlowDirection	X	X	X	X
Focusable	X			
FocusVisualStyle	X			
FontFamily	X	X	X	X
FontSize	X	X	X	X
FontStretch	X	X	X	X
FontStyle	X	X	X	X
FontWeight	X	X	X	X

Property	Area Navigator	Fade Scroller	Marquee	News Scroller
ForceCursor	X			
Foreground	X	X	X	X
Height	X	X	X	X
HorizontalAlign- ment	X	X	X	X
HorizontalContent- Alignment	X	X	X	X
Interval				X
IsEnabled	X	X	X	X
IsHitTestVisible	X	X	X	X
IsManipulationEn- abled	X			
IsOn		X		
IsSharedSizeScope	X			
IsTabStop	X	X	X	X
IsTextSearchCase- Sensitive	X			
IsTextSearchEn- abled	X			
ItemsPanel		X	X	
ItemsSource	X	X	X	
ItemStringFormat	X			
ItemTemplate		X	X	
Language		X	X	X
Localization	X	X	X	X
Margin	X	X	X	X
MarqueeAnima- tionSpeed			X	

Property	Area Navigator	Fade Scroller	Marquee	News Scroller
MarqueeAnimationType			X	
MarqueePauseOnMouseHover			X	
MaxHeight	X	X	X	X
MaxWidth	X	X	X	X
MinHeight	X	X	X	X
MinWidth	X	X	X	X
Name	X	X	X	X
NavigationBrush				X
Opacity	X	X	X	X
OpacityMask	X	X	X	X
OverridesDefaultStyle	X			
Padding	X	X	X	X
Projection		X	X	X
RenderTransform		X	X	X
RenderTransformOrigin	X	X	X	X
RotateDegree	X			
RotateMethod	X			
RotateSpeed	X			
Row	X	X	X	X
RowSpan	X	X	X	X
ShowSliders	X			
SnapsToDevicePixels	X			
Style	X	X	X	X

Property	Area Navigator	Fade Scroller	Marquee	News Scroller
TabIndex	X	X	X	X
TabNavigation		X	X	X
Tag	X	X	X	X
Template	X	X	X	X
TitleBrush				X
TitleFontFamily				X
TitleFontSize				X
TitleFontWeight				X
ToolTip	X	X	X	X
Uid	X	X		X
UseLayoutRounding	X	X	X	X
VerticalAlignment	X	X	X	X
VerticalContentAlignment	X	X	X	X
Visibility	X	X	X	X
Width	X	X	X	X
ZoomLevel	X			

Scrolling Controls Property Descriptions

The following table shows the properties descriptions used by all control:

Property	Description
AllowDrop	Select this check box to indicate that the control can accept data that is dragged onto it.
AlternationCount	Specifies the number of alternating item containers or rows to display.
Background	Expand this field to select the background color of the selected control.

Property	Description
BorderBrush	Expand this field to select the border color of the selected control.
BorderThickness	Specifies the border thickness of a control, in pixels.
CacheMode	Specifies a value that indicates that rendered content should be cached when possible.
Clip	Specifies the geometric shape used to define the outline of the image.
ClipToBounds	Select this check box to clip the content of this control (or content coming from the child elements of this control) to fit into the size of the containing control.
Column	Specifies the column within the grid where the control is located.
ColumnSpan	Specifies the total number of columns that child content spans within a grid.
CornerRadius	Specifies the degree to which the corners of a control are rounded.
Cursor	Specifies the cursor image to display when the mouse pointer hovers over an object.
DataContext	Specifies the data context for a control when it participates in data binding.
DisplayMemberPath	Specifies a path to a value on the source control to serve as the visual representation of the control.
Effect	<p>Click the arrow to expand the property, select the effect you want to apply to the control, and then configure the specific values for the effect. Valid values to select are:</p> <ul style="list-style-type: none"> • Blur <ul style="list-style-type: none"> ◦ Kernel Type ◦ Radius

Property	Description
	<ul style="list-style-type: none"> • Drop Shadow <ul style="list-style-type: none"> ◦ Color ◦ Direction ◦ Opactiy ◦ Shadow Depth
EnableRotation	Select this check box to allow the control to be rotated. If this check box is cleared, the RotateDegree, RotationMethod, and RotationSpeed property configurations will have no effect.
FadeScrollerAnimationSpeed	Specifies the speed at which the control animates.
FadeScrollerAnimationType	<p>Specifies the direction in which the control moves. Valid selections are:</p> <ul style="list-style-type: none"> • Scroll Left Right • Scroll Right Left • Scroll Top Bottom • Scroll Bottom Top • Fade In • None
FadeScrollerDestinationType	<p>Specifies the destination of the scrolling. Valid selections are:</p> <ul style="list-style-type: none"> • Center • Left • Right • Top • Bottom
FadeScrollerFadeInSpeed	Specifies the speed at which the control will fade in.
FadeScrollerFadeOutSpeec	Specifies the speed at which the control will fade out.

Property	Description
FadeScrollerPauseOnMouseHover	Select this check box to indicate whether the control will pause when the mouse pointer is hovered over the control.
FeedRefreshInterval	Specifies the time interval at which the feed content is refreshed from the source specified by the FeedUri property.
FeedUri	Specifies the URI of the news feed whose items should be displayed.
FlowDirection	<p>Specifies the direction that text and other user interface (UI) elements flow within any parent control. Selections are:</p> <ul style="list-style-type: none"> • Left to Right • Right to Left
Focusable	Select this check box to indicate that the control can receive focus.
FocusVisualStyle	Specifies a property that enables customization of appearance, effects, or other style characteristics that apply to this control when it has focus.
FontFamily	Specifies the font family for the control.
FontSize	Specifies the font size for the control.
FontStretch	Specifies the degree to which a font is condensed or expanded on the screen.
FontStyle	Specifies the font style for the control.
FontWeight	Specifies the weight or thickness of the specified font.
ForceCursor	Select this check box to indicate that the user interface (UI) renders the cursor as declared by the Cursor property.
Foreground	Expand this field to select the foreground color.
Height	Specifies the height of the control, in pixels.

Property	Description
HorizontalAlignment	<p>Specifies the horizontal alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
HorizontalContentAlignment	<p>Specifies the horizontal alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
ImageStretch	<p>Specifies how the image should be stretched to fill the destination rectangle. Selections are:</p> <ul style="list-style-type: none"> • Fill • None • Uniform • UniformToFill
Interval	<p>Specifies the amount of time, in milliseconds, between increase or decrease commands when you click a Repeat Button or a Slider.</p>
IsEnabled	<p>Select this check box to indicate that this control is enabled in the user interface (UI).</p>
IsHitTestVisible	<p>Select this check box to indicate that the control can possibly be returned as a hit test result from some portion of its rendered content.</p>
IsManipulationEnabled	<p>Select this check box to indicate that the manipulation events for the control are enabled.</p>
IsOn	<p>Select this check box to indicate that the control is turned on.</p>

Property	Description
IsSharedSizeScope	Select this check box to indicate that controls are sharing size information.
IsTabStop	Select this check box to indicate that a control is included in tab navigation.
IsTextSearchCaseSensitive	Select this check box to indicate that text searches are case sensitive.
IsTextSearchEnabled	Select this check box to indicate that text search is enabled on the control.
ItemsPanel	Specifies the template that defines the panel that controls the layout of items.
ItemsSource	Specifies the collection used to generate the content of the control.
ItemStringFormat	Specifies a composite string that determines how to format the items in the control if they are displayed as strings.
ItemTemplate	Click the ellipsis button to open the Open dialog box to select the data template used to display each item.
Language	Specifies the localization language that applies to an object.
Localization	Click the ellipsis button to open the Localization Configuration dialog box where you can configure the property to localize, the language to use for localization, and the text to localize.
Margin	Specifies the outer margin of a control.
MarqueeAnimationSpeed	Specifies the speed, as a numeric value, of the marquee animation.
MarqueeAnimationType	Specifies the type of marquee animation. Selections are:

Property	Description
	<ul style="list-style-type: none"> • Scroll Left Right • Scroll Right Left • Scroll Top Bottom • Scroll Bottom Top • None
MarqueePauseOnMouseHover	Select this check box to indicate that the marquee pauses when the mouse hovers over it.
MaxHeight	Specifies the maximum height of a control.
MaxWidth	Specifies the maximum width of a control.
MinHeight	Specifies the minimum height of a control.
MinWidth	Specifies the minimum width of a control.
Name	Specifies the identifying name of the control. The name provides a reference so that code behind, such as event handler code, can refer to a markup element during processing by a XAML processor.
NavigationBrush	Expand this field to select the navigation color of the selected control.
Opacity	Specifies the opacity factor applied to the entire control. Valid values are 0 through 1, with up to 15 decimal places.
OpacityMask	Expand this field to select a color for the opacity mask, which is applied to any alpha-channel masking for the rendered content of this control.
OverridesDefaultStyle	Select this check box to indicate that this control incorporates style properties from theme styles.
Padding	Specifies the amount of padding to apply to the inside edge of each side of a control.
Projection	Specifies the 3-D effect to apply to the current object.
RenderTransform	Specifies the transform information that affects the rendering position of an object.

Property	Description
RenderTransformOrigin	<p>Specifies the center point of any possible render transform, relative to the bounds of the element.</p> <ul style="list-style-type: none"> • X is the horizontal origin factor, given as a value between 0 and 1. • Y is the vertical origin factor, given as a value between 0 and 1.
RotateDegree	<p>Specifies the degree the control should be rotated by. For example, if you set the value of this property to 90, the control is rotated 90 degrees.</p>
RotateMethod	<p>Specifies the method to use to rotate the control. Valid selections are:</p> <ul style="list-style-type: none"> • MouseDown • MouseOn
RotationSpeed	<p>Specifies the rotation speed of the control.</p>
Row	<p>RowBackground</p>
RowSpan	<p>Specifies the total number of rows that child content spans within a grid.</p>
ShowSliders	<p>Select this check box to indicate that slider controls will be available within the control's borders.</p>
SnapsToDevicePixels	<p>Select this check box to indicate that rendering for this control uses device-specific pixel settings.</p>
Style	<p>Specifies the style used by this control when it is rendered.</p>
TabIndex	<p>Specifies the order in which controls receive focus when the Tab key is pressed.</p>
TabNavigation	<p>Specifies a value that modifies how tabbing and the TabIndex property work for this control.</p>
Tag	<p>Specifies an arbitrary object value that can be used to store custom information about this control.</p>

Property	Description
Template	Click the ellipsis button to open the Open dialog box to select a template to use.
TitleBrush	Specifies the brush to use to display the title of each article.
TitleFontFamily	Specifies the font family to use to display the title of each article.
TitleFontSize	Specifies the font size for the title.
TitleFontWeight	Specifies the font weight to use to display the title of each article.
ToolTip	Specifies the tool-tip that displays for this control in the user interface (UI).
Uid	Specifies the unique identifier (for localization) for this control.
UseLayoutRounding	Specifies a value that determines whether rendering for the object and its visual subtree should use rounding behavior that aligns rendering to whole pixels.
VerticalAlignment	<p>Specifies the vertical alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
VerticalContentAlignment	<p>Specifies the vertical alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch

Property	Description
Visibility	Specifies the visibility of this control. Selections are: <ul style="list-style-type: none"> • Collapsed • Hidden • Visible
Width	Specifies the width of the control, in pixels.
ZoomLevel	Specifies how much the control will be zoomed in. For example, if you set the value of this property to 10, the control will be zoomed in to 10 times its original size.

Area Navigator



The Area Navigator is a WPF control that is used to contain another control within its borders and allows you manipulate that control by rotating it, zooming in and out, and scrolling.

Use the ItemsSource property to bind to the control you want contained within the Area Navigator control.

See **Scrolling Controls** to view the usage and description of the properties for this control.

Related reference

[Scrolling Controls \(on page 1043\)](#)

Fade Scroller



The Fade Scroller is a Silverlight control that displays its child controls, scrolls through the child controls, and fades in and out as it loops around.

See **Scrolling Controls** to view the usage and description of the properties for this control.

Related reference

[Scrolling Controls \(on page 1043\)](#)

Property	Ellipse	Line	Path	Polygon	Polygon Pencil	Polyline	Polyline Pencil	Rectangle
SnapsTo- Device- Pixels	X	X	X	X	X	X	X	X
Stretch	X		X	X				X
Stroke	X	X	X	X	X	X	X	X
Stroke- DashAr- ray	X	X	X	X	X	X	X	X
Stroke- DashCap	X	X	X	X	X	X	X	X
Strok- DashOff- set	X	X	X	X	X	X	X	X
Stroke- EndLine- Cap	X	X	X	X	X	X	X	X
Stroke- LineJoin	X	X	X	X	X	X	X	X
Stroke- MiterLimit	X	X	X	X	X	X	X	X
StrokeS- tartLine- Cap	X	X	X	X	X	X	X	X
Stroke- Thickness	X	X	X	X	X	X	X	X
Style	X	X	X	X		X	X	X
Tag	X	X	X	X		X	X	X
ToolTip	X	X	X	X		X	X	X
Uid	X	X	X	X		X	X	X

Property	Ellipse	Line	Path	Polygon	Polygon Pencil	Polyline	Polyline Pencil	Rectangle
Use-Layout-Rounding	X	X	X	X		X	X	X
Vertical-Alignment	X	X	X	X		X	X	X
Visibility	X	X	X	X		X	X	X
Width	X	X	X	X		X	X	X
X1		X				X		
X2		X				X		
Y1		X				X		
Y2		X				X		

Shapes Controls Property Descriptions

The following table shows the properties descriptions used by all control:

Property	Description
AllowDrop	Select this check box to indicate that the control can accept data that is dragged onto it.
ClipToBounds	Select this check box to clip the content of this control (or content coming from the child elements of this control) to fit into the size of the containing control.
Column	Specifies the column within the grid where the control is located.
ColumnSpan	Specifies the total number of columns that child content spans within a grid.
Effect	Click the arrow to expand the property, select the effect you want to apply to the control, and then configure the specific values for the effect. Valid values to select are:

Property	Description
	<ul style="list-style-type: none"> • Blur <ul style="list-style-type: none"> ◦ Kernel Type ◦ Radius • Drop Shadow <ul style="list-style-type: none"> ◦ Color ◦ Direction ◦ Opactiy ◦ Shadow Depth
Fill	Expand this field to select the color for the inside of the thumb control or for the interior of the shape.
FlowDirection	<p>Specifies the direction that text and other user interface (UI) elements flow within any parent control. Selections are:</p> <ul style="list-style-type: none"> • Left to Right • Right to Left
Focusable	Select this check box to indicate that the control can receive focus.
FocusVisualStyle	Specifies a property that enables customization of appearance, effects, or other style characteristics that apply to this control when it has focus.
ForceCursor	Select this check box to indicate that the user interface (UI) renders the cursor as declared by the Cursor property.
Height	Specifies the height of the control, in pixels.
HorizontalAlignment	<p>Specifies the horizontal alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch

Property	Description
IsEnabled	Select this check box to indicate that this control is enabled in the user interface (UI).
IsHitTestVisible	Select this check box to indicate that the control can possibly be returned as a hit test result from some portion of its rendered content.
IsManipulationEnabled	Select this check box to indicate that the manipulation events for the control are enabled.
IsSharedSizeScope	Select this check box to indicate that controls are sharing size information.
Localization	Click the ellipsis button to open the Localization Configuration dialog box where you can configure the property to localize, the language to use for localization, and the text to localize.
Margin	Specifies the outer margin of a control.
MaxHeight	Specifies the maximum height of a control.
MaxWidth	Specifies the maximum width of a control.
MinHeight	Specifies the minimum height of a control.
MinWidth	Specifies the minimum width of a control.
Name	Specifies the identifying name of the control. The name provides a reference so that code behind, such as event handler code, can refer to a markup element during processing by a XAML processor.
Opacity	Specifies the opacity factor applied to the entire control. Valid values are 0 through 1, with up to 15 decimal places.
OpacityMask	Expand this field to select a color for the opacity mask, which is applied to any alpha-channel masking for the rendered content of this control.
OverridesDefaultStyle	Select this check box to indicate that this control incorporates style properties from theme styles.

Property	Description
RadiusX	Specifies the x-axis radius of the ellipse that is used to round the corners of the rectangle.
RadiusY	Specifies the y-axis radius of the ellipse that is used to round the corners of the rectangle.
ReloadableForm	<p>Select this check box to enable viewing of this form in a completed or acquired state in the Task List. This property works in conjunction with workflow Tracking Configuration, which must be enabled to view completed forms.</p> <div data-bbox="821 737 1422 1136" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note: If you enable this setting, you must configure your form to handle viewing in a completed or acquired (reloaded) state by using the IsReloaded expression variable. For example, if the form is in a reloaded state, you can disable certain user interface elements and show read-only data.</p> </div>
Render Transform	Specifies the transform information that affects the rendering position of an object.
RenderTransformOrigin	<p>Specifies the center point of any possible render transform, relative to the bounds of the element.</p> <ul style="list-style-type: none"> • X is the horizontal origin factor, given as a value between 0 and 1. • Y is the vertical origin factor, given as a value between 0 and 1.
Row	RowBackground
RowSpan	Specifies the total number of rows that child content spans within a grid.
SnapsToDevicePixels	Select this check box to indicate that rendering for this control uses device-specific pixel settings.

Property	Description
Stretch	<p>Specifies how an Image stretches to fill the destination rectangle. Selections are:</p> <ul style="list-style-type: none"> • Fill • None (does not apply to Scale Box) • Uniform (does not apply to Scale Box) • UniformFill (does not apply to Scale Box)
Stroke	<p>Expand this field to select the color for the shape's outline.</p>
StrokeDashArray	<p>Click the ellipsis button to open the StrokeDash-Array Property Editor dialog box to indicate the pattern of dashes and gaps that is used to outline shapes.</p>
StrokeDashCap	<p>Specifies how the ends of a dash are drawn. Selections are:</p> <ul style="list-style-type: none"> • Flat • Round • Square • Triangle
StrokeDashOffset	<p>Specifies the distance within the dash pattern where a dash begins</p>
StrokeEndLineCap	<p>Specifies the shape at the end of a line. Selections are:</p> <ul style="list-style-type: none"> • Flat • Round • Square • Triangle
StrokeLineJoin	<p>Specifies the type of join that is used at the vertices of a shape. Selection are:</p>

Property	Description
	<ul style="list-style-type: none"> • Bevel • Miter • Round
StrokeMiterLimit	Specifies a limit on the ratio of the miter length to half the StrokeThickness of a shape.
StrokeStartLineCap	<p>Specifies the shape at the start of a line. Selections are:</p> <ul style="list-style-type: none"> • Flat • Round • Square • Triangle
StrokeThickness	Specifies the width of the control's outline.
Style	Specifies the style used by this control when it is rendered.
Tag	Specifies an arbitrary object value that can be used to store custom information about this control.
ToolTip	Specifies the tool-tip that displays for this control in the user interface (UI).
Uid	Specifies the unique identifier (for localization) for this control.
UseLayoutRounding	Specifies a value that determines whether rendering for the object and its visual subtree should use rounding behavior that aligns rendering to whole pixels.
VerticalAlignment	<p>Specifies the vertical alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch

Property	Description
Visibility	Specifies the visibility of this control. Selections are: <ul style="list-style-type: none"> • Collapsed • Hidden • Visible
Width	Specifies the width of the control, in pixels.
X1	Specifies the position of the line's starting point on the X axis.
X2	Specifies the position of the line's ending point on the X axis.
Y1	Specifies the position of the line's starting point on the Y axis.
Y2	Specifies the position of the line's ending point on the Y axis.

Configure a path control

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. Expand the control group you want to work with.
6. Click and drag the control to the desired position in the **Designer**.



Note:

To apply the properties of a control to another control, click and drag the control to another control in the document tree. The Applying **<control (control name)>** to **Target** dialog box appears where you can select the option you want to use to apply to the control.

7. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

8. Resize the path control using the resize handles, as required.

9. Right-click the path control and select **Edit Layer**.

The control's outline color changes to orange to indicate that it is the active layer, and the contents of the Toolbox tab changes to display the list of Geometry shapes.

10. Click the geometry shape you want to add to the path control, and then move the mouse to the active layer area.

The mouse pointer changes to a cross-hair.

11. Depending on the geometry shape you are adding, you can click and drag the mouse in the active layer to create the shape you want, or click multiple times to add start and end points for the shape you want to create.



Tip:

- To clear the shape while you are still designing it, right-click the shape, and then press the delete button.
- To clear a previously created shape, click the shape, and then press the delete button.

12. To close the active layer, click **Close Active Geometry**, and then click in the active layer.

13. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Related reference

[Shapes \(on page 1057\)](#)

Related information

[Designer: Control Toolbox \(on page 725\)](#)

Add an ellipse

The ellipse control creates a circular shape.

Before you begin

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. From the Shapes list, select the ellipse control.



Note:

In WPF forms, the mouse pointer changes to a cross-hair.

6. In the designer, click and drag the mouse to draw the ellipse shape you want.
7. When the ellipse is complete, release the mouse to stop drawing.



Note:

Any time you select the shape, each of the points is visible. You can then click and drag one point at a time to reshape the drawing.

8. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Related reference

[Shapes \(on page 1057\)](#)

Related information

[Designer: Control Toolbox \(on page 725\)](#)

Add a line

Shapes are typically used on a canvas, but can be used in other containers as well.

Before you begin**Procedure**

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. From the Shapes list, select the line control.

**Note:**

In WPF forms, the mouse pointer changes to a cross-hair.

6. In the designer, click where you want the line to start, drag the mouse pointer to the end position, and then release the mouse.

**Note:**

Any time you select the shape, each of the points is visible. You can then click and drag one point at a time to position the line.

7. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Related reference

[Shapes \(on page 1057\)](#)

Related information

[Add a polyline \(on page 1072\)](#)

Add a polygon

The polygon control allows you to draw multi-sided shapes.

Before you begin**Procedure**

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. From the Shapes list, select the polygon control.

**Note:**

In WPF forms, the mouse pointer changes to a cross-hair.

6. In the designer, click and release the mouse for each point of the polygon. Each point is connected to the previous point; therefore, after each click, a line appears between the points.
7. When the polygon is complete, right-click to stop adding points to it.

**Note:**

Any time you select the shape, each of the points is visible. You can then click and drag one point at a time to reshape the drawing.

8. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Related reference

[Shapes \(on page 1057\)](#)

Related information

[Add a polygon pencil \(on page 1071\)](#)

[Designer: Control Toolbox \(on page 725\)](#)

Add a polygon pencil

The polygon pencil control allows you to draw freeform shapes.

Before you begin**Procedure**

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. From the Shapes list, select the polygon pencil control.

**Note:**

In WPF forms, the mouse pointer changes to a cross-hair.

6. In the designer, click and drag the mouse to draw the shape you want.
7. When the polygon is complete, release the mouse to stop drawing.

**Note:**

Any time you select the shape, each of the points is visible. You can then click and drag one point at a time to reshape the drawing.

8. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Related reference

[Shapes \(on page 1057\)](#)

Related information

[Add a polygon \(on page 1070\)](#)

[Add a polyline pencil \(on page 1073\)](#)

[Designer: Control Toolbox \(on page 725\)](#)

Add a polyline

The polyline control functions like the polygon, but it does not automatically close the shape for you.

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. From the Shapes list, select the polyline control.

**Note:**

In WPF forms, the mouse pointer changes to a cross-hair.

6. In the designer, click and release the mouse for each point of the polyline. Each point is connected to the previous point; therefore, after each click, a line appears between the points.
7. When the polyline is complete, right-click to stop adding points to it.

**Note:**

Any time you select the shape, each of the points is visible. You can then click and drag one point at a time to reshape the drawing.

8. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Related reference

[Shapes \(on page 1057\)](#)

Related information

[Add a polyline pencil \(on page 1073\)](#)

[Designer: Control Toolbox \(on page 725\)](#)

Add a polyline pencil

The polyline pencil control functions like the polygon pencil, but it does not automatically close the shape for you.

Before you begin**Procedure**

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. From the Shapes list, select the polyline pencil control.

**Note:**

In WPF forms, the mouse pointer changes to a cross-hair.

6. In the designer, click and drag the cross-hair to draw the shape you want.
7. When the polyline is complete, release the mouse to stop drawing.

**Note:**

Any time you select the shape, each of the points is visible. You can then click and drag one point at a time to reshape the drawing.

8. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Related reference

[Shapes \(on page 1057\)](#)

Related information

[Add a polygon pencil \(on page 1071\)](#)

[Designer: Control Toolbox \(on page 725\)](#)

Add a rectangle

The rectangle control adds a rectangular shape to the designer.

About this task

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. From the Shapes list, select the polyline pencil control.



Note:

In WPF forms, the mouse pointer changes to a cross-hair.

6. In the designer, click and drag the cross-hair to draw the shape you want.
7. When the rectangle is complete, release the mouse to stop drawing.



Note:

Any time you select the shape, each of the points is visible. You can then click and drag one point at a time to reshape the drawing.

8. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Related reference

[Shapes \(on page 1057\)](#)

Related information

[Designer: Control Toolbox \(on page 725\)](#)

Ellipse



The Ellipse control draws a simple ellipse shape.

See **Shape Controls** to view the usage and description of the properties for this control.

Related reference

[Shapes \(on page 1057\)](#)

Line



The Line control is a graphical control that displays a horizontal, vertical, or diagonal line that cannot be changed directly.

Although lines have a Fill property that is similar to the Fill property found in a rectangle or an ellipse, changing the Fill property of a line does not change to the line's appearance as it does for a rectangle or an ellipse. Because a line is essentially a linear stroke between two end points, you cannot connect these two points together to form another shape.

See **Shape Controls** to view the usage and description of the properties for this control.

Related reference

[Shapes \(on page 1057\)](#)

Path



A Path control is a two-dimensional graphic derived from the Shape class. It draws a series of connected lines and curves. Paths can be very simple or very complex forming elaborate images and diagrams.

When you add a path control, a data set of geometry shapes displays on the Toolbox tab. You can add any combination of geometry shapes to the path control to create an image or design.

See **Shape Controls** to view the usage and description of the properties for this control.

Related reference

[Shapes \(on page 1057\)](#)

Polygon



The Polygon control is a shape defined by a series of end points and linear lines that is useful for drawings, graphic designs, and user interfaces.

A polygon can also consist of an infinite number of end points. The main difference between a polygon and a polyline is that a polygon, after it has been drawn, closes itself off into a complete shape between its start and end points.

See **Shape Controls** to view the usage and description of the properties for this control.

Related reference

[Shapes \(on page 1057\)](#)

Polygon Pencil



The Polygon Pencil control allows you to draw a series of points connected to one another as dictated by your hand movements.

Rather than drawing the connections at specified points in straight lines, the polygon pencil adds points to the shape as the mouse moves. The polygon objects produced by using the pencil are the same as the objects used by the regular polygon drawing methods, they just contain more points.

See **Shape Controls** to view the usage and description of the properties for this control.

Related reference

[Shapes \(on page 1057\)](#)

Polyline



The Polyline control is used to create drawings, graphic designs, and user interfaces. A polyline draws linear (that is, straight) lines by connecting a number of specified points. Whereas a simple line control contains only one line between two end points, a polyline can contain an infinite number of points.

Unlike a rectangle or an ellipse, a polyline is drawn as an unfinished shape, meaning that the starting and ending points of a polyline are left open and unconnected. But, similar to other finished shapes, a polyline is capable of displaying a fill color or design on the inside of the connected lines and end points.

See **Shape Controls** to view the usage and description of the properties for this control.

Related reference

[Shapes \(on page 1057\)](#)

Polyline Pencil



The Polyline Pencil control allows you to draw a series of points connected to one another as dictated by your hand movements.

Rather than drawing the connections at specified points in straight lines, the polyline pencil adds points to the shape as the mouse moves. The polyline objects produced by using the pencil are the same as the objects used by the regular polyline drawing method, they just contain more points.

The polyline pencil has more flexibility than a polyline because it behaves more like a drawing tool and looks more like a series of points connected by a line, rather than a line connected by two end points.

The polyline pencil is used to draw on a canvas control that is set as the active layer.

See **Shape Controls** to view the usage and description of the properties for this control.

Related reference

[Shapes \(on page 1057\)](#)

Rectangle



The Rectangle control draws a basic rectangle shape.

The rectangle has RadiusX and RadiusY properties that can be used to curve its corners.

See **Shape Controls** to view the usage and description of the properties for this control.

Related reference

[Shapes \(on page 1057\)](#)

Slider Controls

Sliders allow you to graphically manipulate the value of the control within a finite range by moving a thumb or value indicator along a track.

The incremental readout on a slider can be on one or both sides. When a slider control receives a value outside the specified range (that is, above the Maximum or below the Minimum), or the value is Null, the control sets its Opacity property to 0.2. The control appears disabled until the value is back within the specified range or is no longer Null.

Slider Controls Property Usage

The following table shows the properties used by each control:

Property	Horizontal Slider	Vertical Slider
AllowDrop	X	X
Background	X	X
BorderBrush	X	X
BorderThickness	X	X
ClipToBounds	X	X
Column	X	X
ColumnSpan	X	X
CornerRadius	X	X
Effect	X	X
Fill	X	X
FlowDirection	X	X
Focusable	X	X
FocusVisualStyle	X	X
FontFamily	X	X
FontSize	X	X
FontStretch	X	X
FontStyle	X	X
FontWeight	X	X
ForceCursor	X	X
Foreground	X	X
GlareBrush	X	X
Height	X	X
HorizontalAlignment	X	X
HorizontalContentAlignment	X	X

Property	Horizontal Slider	Vertical Slider
InnerBorderBrush	X	X
InnerBorderThickness	X	X
InnerCornerRadius	X	X
IsAnnotated	X	X
IsEnabled	X	X
IsHitTestVisible	X	X
IsManipulationEnabled	X	X
IsSharedSizeScope	X	X
IsTabStop	X	X
Localization	X	X
Margin	X	X
MaxHeight	X	X
Maximum	X	X
MaxWidth	X	X
MinHeight	X	X
Minimum	X	X
MinWidth	X	X
Name	X	X
Opacity	X	X
OpacityMask	X	X
OverridesDefaultStyle	X	X
Padding	X	X
RenderTransformOrigin	X	X
Row	X	X
RowSpan	X	X
SnapsToDevicePixels	X	X

Property	Horizontal Slider	Vertical Slider
StrokeThickness		
Style	X	X
TabIndex	X	X
Tag	X	X
Template	X	X
ThumbStyle	X	X
TickFrequency	X	X
Ticks	X	X
ToolTip	X	X
Uid	X	X
UseLayoutRounding	X	X
Value	X	X
VerticalAlignment	X	X
VerticalContentAlignment	X	X
Visibility	X	X
Width	X	X

Slider Controls Property Descriptions

The following table shows the properties descriptions used by all control:

Property	Description
AllowDrop	Select this check box to indicate that the control can accept data that is dragged onto it.
Background	Expand this field to select the background color of the selected control.
BorderBrush	Expand this field to select the border color of the selected control.

Property	Description
BorderThickness	Specifies the border thickness of a control, in pixels.
ClipToBounds	Select this check box to clip the content of this control (or content coming from the child elements of this control) to fit into the size of the containing control.
Column	Specifies the column within the grid where the control is located.
ColumnSpan	Specifies the total number of columns that child content spans within a grid.
CornerRadius	Specifies the degree to which the corners of a control are rounded.
Effect	<p>Click the arrow to expand the property, select the effect you want to apply to the control, and then configure the specific values for the effect. Valid values to select are:</p> <ul style="list-style-type: none"> • Blur <ul style="list-style-type: none"> ◦ Kernel Type ◦ Radius • Drop Shadow <ul style="list-style-type: none"> ◦ Color ◦ Direction ◦ Opacity ◦ Shadow Depth
Fill	Expand this field to select the color for the inside of the thumb control or for the interior of the shape.
FlowDirection	<p>Specifies the direction that text and other user interface (UI) elements flow within any parent control. Selections are:</p> <ul style="list-style-type: none"> • Left to Right • Right to Left

Property	Description
Focusable	Select this check box to indicate that the control can receive focus.
FocusVisualStyle	Specifies a property that enables customization of appearance, effects, or other style characteristics that apply to this control when it has focus.
FontFamily	Specifies the font family for the control.
FontSize	Specifies the font size for the control.
FontStretch	Specifies the degree to which a font is condensed or expanded on the screen.
FontStyle	Specifies the font style for the control.
FontWeight	Specifies the weight or thickness of the specified font.
ForceCursor	Select this check box to indicate that the user interface (UI) renders the cursor as declared by the Cursor property.
Foreground	Expand this field to select the foreground color.
GlareBrush	Expand this field to specify the glare effect for the control.
Height	Specifies the height of the control, in pixels.
HorizontalAlignment	<p>Specifies the horizontal alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
HorizontalContentAlignment	<p>Specifies the horizontal alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Left • Center

Property	Description
	<ul style="list-style-type: none"> • Right • Stretch
ImageStretch	<p>Specifies how the image should be stretched to fill the destination rectangle. Selections are:</p> <ul style="list-style-type: none"> • Fill • None • Uniform • UniformToFill
InnerBorderBrush	Expand this field to select the color for the inner border.
InnerBorderThickness	Expand this field to select the color for the inner border.
InnerCornerRadius	Specifies the degree to which the corners of the inner border are rounded.
IsAnnotated	Select this check box to indicate that value labels display with the ticks.
IsEnabled	Select this check box to indicate that this control is enabled in the user interface (UI).
IsHitTestVisible	Select this check box to indicate that the control can possibly be returned as a hit test result from some portion of its rendered content.
IsManipulationEnabled	Select this check box to indicate that the manipulation events for the control are enabled.
IsSharedSizeScope	Select this check box to indicate that controls are sharing size information.
IsTabStop	Select this check box to indicate that a control is included in tab navigation.
Localization	Click the ellipsis button to open the Localization Configuration dialog box where you can configure the property to localize, the language to use for localization, and the text to localize.

Property	Description
Margin	Specifies the outer margin of a control.
MaxHeight	Specifies the maximum height of a control.
Maximum	Specifies the maximum value of the range of the control.
MaxWidth	Specifies the maximum width of a control.
MinHeight	Specifies the minimum height of a control.
Minimum	Specifies the minimum value of the range of the control.
MinWidth	Specifies the minimum width of a control.
Name	Specifies the identifying name of the control. The name provides a reference so that code behind, such as event handler code, can refer to a markup element during processing by a XAML processor.
Opacity	Specifies the opacity factor applied to the entire control. Valid values are 0 through 1, with up to 15 decimal places.
OpacityMask	Expand this field to select a color for the opacity mask, which is applied to any alpha-channel masking for the rendered content of this control.
OverridesDefaultStyle	Select this check box to indicate that this control incorporates style properties from theme styles.
Padding	Specifies the amount of padding to apply to the inside edge of each side of a control.
RenderTransformOrigin	<p>Specifies the center point of any possible render transform, relative to the bounds of the element.</p> <ul style="list-style-type: none"> • X is the horizontal origin factor, given as a value between 0 and 1. • Y is the vertical origin factor, given as a value between 0 and 1.
Row	RowBackground

Property	Description
RowSpan	Specifies the total number of rows that child content spans within a grid.
SnapsToDevicePixels	Select this check box to indicate that rendering for this control uses device-specific pixel settings.
StrokeThickness	Specifies the width of the control's outline.
Style	Specifies the style used by this control when it is rendered.
TabIndex	Specifies the order in which controls receive focus when the Tab key is pressed.
Tag	Specifies an arbitrary object value that can be used to store custom information about this control.
Template	Click the ellipsis button to open the Open dialog box to select a template to use.
ThumbStyle	<p>Specifies the shape used for the thumb control. Selections are:</p> <ul style="list-style-type: none"> • Bar • Diamond • Ellipse • Knob • Pentagon • Triangle
TickFrequency	Specifies the interval between tick marks.
Ticks	For linear gauges and sliders, click the ellipsis button to open the Tick Property Editor dialog box to enter the marks to display for a Slider.
ToolTip	Specifies the tool-tip that displays for this control in the user interface (UI).
Uid	Specifies the unique identifier (for localization) for this control.

Property	Description
UseLayoutRounding	Specifies a value that determines whether rendering for the object and its visual subtree should use rounding behavior that aligns rendering to whole pixels.
Value	Specifies the current magnitude of the range control.
VerticalAlignment	Specifies the vertical alignment characteristics applied to this control. Selections are: <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
VerticalContentAlignment	Specifies the vertical alignment of the control's content. Selections are: <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
Visibility	Specifies the visibility of this control. Selections are: <ul style="list-style-type: none"> • Collapsed • Hidden • Visible
Width	Specifies the width of the control, in pixels.

Horizontal Slider

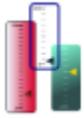


The Horizontal Slider control allows you to adjust a finite numeric value using a configurable Thumb Style.

See **Slider Controls** to view the usage and description of the properties for this control.

Related reference[Slider Controls \(on page 1078\)](#)

Vertical Slider



The Vertical Slider control allows you to adjust a finite numeric value using a configurable Thumb Style.

See **Slider Controls** to view the usage and description of the properties for this control.

Related reference[Slider Controls \(on page 1078\)](#)

Viewer Controls

The Designer provides the following group of custom viewer controls.

Viewer Controls Property Usage

The following table shows the properties used by each control:

Property	Scroll Viewer
AllowDrop	X
Background	X
BorderBrush	X
BorderThickness	X
CanContentScroll	X
ClipToBounds	X
Column	X
ColumnSpan	X
Content	X
ContentStringFormat	X
Effect	X

Property	Scroll Viewer
FlowDirection	X
Focusable	X
FocusVisualStyle	X
FontFamily	X
FontSize	X
FontStretch	X
FontStyle	X
FontWeight	X
ForceCursor	X
Foreground	X
Height	X
HorizontalAlignment	X
HorizontalContentAlignment	X
HorizontalScrollBarVisibility	X
IsDeferredScrollingEnabled	X
IsEnabled	X
IsHitTestVisible	X
IsManipulationEnabled	X
IsSharedSizeScope	X
IsTabStop	X
Localization	X
Margin	X
MaxHeight	X
MaxWidth	X
MinHeight	X
MinWidth	X

Property	Scroll Viewer
Name	X
Opacity	X
OpacityMask	X
OverridesDefaultStyle	X
Padding	X
PanningDeclaration	X
PanningMode	X
PanningRatio	X
RenderTransformOrigin	X
Row	X
RowSpan	X
SnapsToDevicePixels	X
StrokeThickness	X
Style	X
TabIndex	X
Tag	X
Template	X
ToolTip	X
Uid	X
UseLayoutRounding	X
VerticalAlignment	X
VerticalContentAlignment	X
VerticalScrollBarVisibility	X
Visibility	X
Width	X

Viewer Controls Property Descriptions

The following table shows the properties descriptions used by all control:

Property	Description
AllowDrop	Select this check box to indicate that the control can accept data that is dragged onto it.
Background	Expand this field to select the background color of the selected control.
BorderBrush	Expand this field to select the border color of the selected control.
BorderThickness	Specifies the border thickness of a control, in pixels.
CanContentScroll	Select this check box to indicate that elements that support the Scroll Viewer control are allowed to scroll.
ClipToBounds	Select this check box to clip the content of this control (or content coming from the child elements of this control) to fit into the size of the containing control.
Column	Specifies the column within the grid where the control is located.
ColumnSpan	Specifies the total number of columns that child content spans within a grid.
Content	Specifies the content displayed on the control.
ContentStringFormat	Specifies a composite string that determines how to format the Content property if it is displayed as a string.
Effect	Click the arrow to expand the property, select the effect you want to apply to the control, and then configure the specific values for the effect. Valid values to select are:

Property	Description
	<ul style="list-style-type: none"> • Blur <ul style="list-style-type: none"> ◦ Kernel Type ◦ Radius • Drop Shadow <ul style="list-style-type: none"> ◦ Color ◦ Direction ◦ Opactiy ◦ Shadow Depth
FlowDirection	<p>Specifies the direction that text and other user interface (UI) elements flow within any parent control. Selections are:</p> <ul style="list-style-type: none"> • Left to Right • Right to Left
Focusable	<p>Select this check box to indicate that the control can receive focus.</p>
FocusVisualStyle	<p>Specifies a property that enables customization of appearance, effects, or other style characteristics that apply to this control when it has focus.</p>
FontFamily	<p>Specifies the font family for the control.</p>
FontSize	<p>Specifies the font size for the control.</p>
FontStretch	<p>Specifies the degree to which a font is condensed or expanded on the screen.</p>
FontStyle	<p>Specifies the font style for the control.</p>
FontWeight	<p>Specifies the weight or thickness of the specified font.</p>
ForceCursor	<p>Select this check box to indicate that the user interface (UI) renders the cursor as declared by the Cursor property.</p>
Foreground	<p>Expand this field to select the foreground color.</p>
Height	<p>Specifies the height of the control, in pixels.</p>

Property	Description
HorizontalAlignment	<p>Specifies the horizontal alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
HorizontalContentAlignment	<p>Specifies the horizontal alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Left • Center • Right • Stretch
HorizontalScrollBarVisibility	<p>Specifies whether a horizontal scroll bar is shown. Selections are:</p> <ul style="list-style-type: none"> • Auto - the scroll bar appears only if the content exceeds the width of the control • Disabled - the scroll bar never appears • Hidden - the scroll bar is hidden from view • Visible - the scroll bar is visible at all times
ImageStretch	<p>Specifies how the image should be stretched to fill the destination rectangle. Selections are:</p> <ul style="list-style-type: none"> • Fill • None • Uniform • UniformToFill
IsDeferredScrollingEnabled	<p>Select this check box to indicate that the content is stationary when you drag the thumb of a scroll bar.</p>
IsEnabled	<p>Select this check box to indicate that this control is enabled in the user interface (UI).</p>

Property	Description
IsHitTestVisible	Select this check box to indicate that the control can possibly be returned as a hit test result from some portion of its rendered content.
IsManipulationEnabled	Select this check box to indicate that the manipulation events for the control are enabled.
IsSharedSizeScope	Select this check box to indicate that controls are sharing size information.
IsTabStop	Select this check box to indicate that a control is included in tab navigation.
Localization	Click the ellipsis button to open the Localization Configuration dialog box where you can configure the property to localize, the language to use for localization, and the text to localize.
Margin	Specifies the outer margin of a control.
MaxHeight	Specifies the maximum height of a control.
MaxWidth	Specifies the maximum width of a control.
MinHeight	Specifies the minimum height of a control.
MinWidth	Specifies the minimum width of a control.
Name	Specifies the identifying name of the control. The name provides a reference so that code behind, such as event handler code, can refer to a markup element during processing by a XAML processor.
Opacity	Specifies the opacity factor applied to the entire control. Valid values are 0 through 1, with up to 15 decimal places.
OpacityMask	Expand this field to select a color for the opacity mask, which is applied to any alpha-channel masking for the rendered content of this control.
OverridesDefaultStyle	Select this check box to indicate that this control incorporates style properties from theme styles.

Property	Description
Padding	Specifies the amount of padding to apply to the inside edge of each side of a control.
PanningDeceleration	Specifies the rate at which the control slows.
PanningMode	<p>Specifies the way the control reacts to touch manipulation. Valid selections are:</p> <ul style="list-style-type: none"> • None • Horizontal Only • Vertical Only • Both • Horizontal First • Vertical First
PanningRatio	Specifies the ratio of scrolling offset to translate manipulation offset.
RenderTransformOrigin	<p>Specifies the center point of any possible render transform, relative to the bounds of the element.</p> <ul style="list-style-type: none"> • X is the horizontal origin factor, given as a value between 0 and 1. • Y is the vertical origin factor, given as a value between 0 and 1.
Row	RowBackground
RowSpan	Specifies the total number of rows that child content spans within a grid.
SnapsToDevicePixels	Select this check box to indicate that rendering for this control uses device-specific pixel settings.
StrokeThickness	Specifies the width of the control's outline.
Style	Specifies the style used by this control when it is rendered.
TabIndex	Specifies the order in which controls receive focus when the Tab key is pressed.

Property	Description
Tag	Specifies an arbitrary object value that can be used to store custom information about this control.
Template	Click the ellipsis button to open the Open dialog box to select a template to use.
ToolTip	Specifies the tool-tip that displays for this control in the user interface (UI).
Uid	Specifies the unique identifier (for localization) for this control.
UseLayoutRounding	Specifies a value that determines whether rendering for the object and its visual subtree should use rounding behavior that aligns rendering to whole pixels.
VerticalAlignment	<p>Specifies the vertical alignment characteristics applied to this control. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
VerticalContentAlignment	<p>Specifies the vertical alignment of the control's content. Selections are:</p> <ul style="list-style-type: none"> • Bottom • Center • Top • Stretch
VerticalScrollBarVisibility	<p>Specifies whether a vertical scroll bar is shown. Scroll bar selections are:</p> <ul style="list-style-type: none"> • Auto - appears only if the content exceeds the width of the control • Disabled - never appears

Property	Description
	<ul style="list-style-type: none"> • Hidden - hidden from view • Visible - visible at all times.
Visibility	<p>Specifies the visibility of this control. Selections are:</p> <ul style="list-style-type: none"> • Collapsed • Hidden • Visible
Width	Specifies the width of the control, in pixels.

Add content to a scroll viewer

The Scroll Viewer control encapsulates horizontal and vertical scroll bar elements and a content container, such as a grid or panel, in order to display other visible controls in a scrollable area.

Before you begin

Procedure

1. Add a scroll viewer to the designer.
2. Add content to the designer (for example, in a text block).
3. Select the content, and then right-click and select **Cut**.
4. Select the scroll viewer, and then right-click and select **Paste Special**.
The Apply Item to Scrollviewer dialog box appears.
5. In a WPF form, select **Child of ScrollViewer**. In a Silverlight form, select **Content**.
6. Click **OK**.
7. In the Document Tree, select the object you just pasted (for example, the text block).
8. Click the **Properties** tab, and then select the Advanced check box.
9. In the Margin, Height, and Width properties, click the arrow and select **Reset to Default**.
10. In the Text Wrapping property, click **Wrap**.
11. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Scroll Viewer



The Scroll Viewer control provides a way to enable scrolling in a form or user display.

The Scroll Viewer control encapsulates horizontal and vertical scroll bar elements and a content container, such as a grid or panel, in order to display other visible controls in a scrollable area.

A scroll viewer can have only one child control, typically a grid or panel control, that can contain a collection of child controls. The Content property defines the child control of the scroll viewer.

See **Viewer Controls** to view the usage and description of the properties for this control.

Related reference

[Viewer Controls \(on page 1088\)](#)

Designer: Resources

The Resources tab contains images that you can use in your forms or user displays.

The following table lists the types of images available to use.

Type	Description
Air Units	Motors
Boiler	Pumps
Brewing Equipment	Shipping Material
Buildings	Signs
Conveyors	Switches
Cooling Equipment	Symbols
Fans	Tanks
Heating Equipment	Tools
Icons	Towers
Machines	Transportation
Miscellaneous Equipment	Valves Pipes

Designer: Properties Panel

Each control available in the Toolbox has properties that you can configure. These properties can be used to change the appearance of the control, as well as certain functionality.

Each control has a set of basic properties, as well as more advanced properties that are displayed by selecting the Advanced check box. There are properties that are common to all of the controls, properties that are common to types of controls, and some controls have properties that are exclusive.

For more information on properties, see www.msdn.com

In addition to displaying advanced properties, the Properties panel provides the ability to sort the properties in different ways.

The following table describes the sort capabilities:

Sort Button	Sorts the properties...
Alphabetical	in ascending alphabetical order.
In Use	to display only the ones in use.
By Category	according to assigned categories.
By Type	according to assigned type.
By Class	according to assigned class.

Properties can be configured by using the visual controls, such as entering text, selecting a check box, or clicking a button. You can also choose to configure properties by using a bind or expression to define the input parameters.

Navigate URI

The Navigate URI dialog box is used throughout the Designer to select source files to use within a form or user display.

You can select source files from the local Resources directory in the Project panel, or you can select source files from a remote location. A remote location can be a location on your local computer or on your network.

The following table describes the properties found on the Navigate URI dialog box.

Property	Value	Description
URI address	User-defined	<p>Specifies the source file to use for the Image control.</p> <p>If the From Resources option is selected, you can select the source file from the drop-down list.</p> <p>If the From URI option is selected, you must use the ellipsis button to open the Select File dialog box where you can browse for and select the source file you want to use. If you use this option, the absolute path for the source file location displays in this field.</p>
Ellipsis	N/A	Click this button to browse for a source file from the selected location. This button is available only when the From URI option is selected.
From Project Resources	N/A	<p>Select this option to select a source file from the URI address drop-down list. The source files available in this list must have been added to the Resources directory in the Project panel.</p> <div data-bbox="1024 1556 1425 1887" style="border: 1px solid #ccc; border-radius: 10px; background-color: #fff9c4; padding: 10px; margin-top: 10px;"> <p> Important:</p> <p>When an image file is added to the project (that is, in the Resources folder) a copy of that image file is made. The copy then stays with</p> </div>

Property	Value	Description
		<p> the project and can be viewed on any Client machine that the project is opened on.</p>
From URI	N/A	<p>Select this option to select a source file from a remote location.</p> <p> Important:</p> <p>If you use a remote image file in your project, it must be stored in a network share location. If you use an image that is located on your local machine, only the absolute path is maintained. Other Client machines will not be able to display the image.</p> <p>For example, if you point to C:\MyPics\image.jpg, then all client machines must have the image stored in the same location. However, if the image is stored at \\<network share>\image.jpg, and you point to that location, then all client machines on your network are able to display the image.</p>

Add a local source file

Before you begin

You must have added image source files to the Resources directory on the Project panel in order to select a local source file.

Procedure

1. In the navigator, click **Global Displays**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. Expand the control group you want to work with.
6. Click and drag the control to the desired position in the **Designer**.



Note:

To apply the properties of a control to another control, click and drag the control to another control in the document tree. The Applying **<control (control name)>** to **Target** dialog box appears where you can select the option you want to use to apply to the control.

7. Click the **Properties** tab.
8. Click the ellipsis button next to the **Source** property.

The **Navigate Uri** dialog box appears.

9. Select the **From Project Resources** option, and then from the URI address drop-down list, select the source file you want to use.
10. Click **OK**.
11. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Add a remote source file

Before you begin

Procedure

1. In the navigator, click **Global Displays**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Designer**.
4. Click the **Toolbox** tab.
5. Expand the control group you want to work with.
6. Click and drag the control to the desired position in the **Designer**.



Note:

To apply the properties of a control to another control, click and drag the control to another control in the document tree. The Applying **<control (control name)>** to **Target** dialog box appears where you can select the option you want to use to apply to the control.

7. Click the **Properties** tab.
8. Click the ellipsis button next to the **Source** property.

The **Navigate Uri** dialog box appears.

9. Select the **From URI** option, and then click the ellipsis button.

The Select File dialog box appears.

10. Locate and select the source file you want to use, and then click **Open**.
11. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Designer: Parameters Panel

The Parameters panel allows you to configure input and output parameters that you can use to bind control properties to.

The following table describes the attributes of this panel:

Attribute	Value	Description
Input Parameters	User-defined	Displays the list of input parameters.
Input Parameters: Add	N/A	Click to add an input parameter.
Input Parameters: Remove	N/A	Click to remove the selected input parameter.
Output Parameters	User-defined	Displays the list of output parameters.
Output Parameters: Add	N/A	Click to add an output parameter.
Output Parameters: Remove	N/A	Click to remove the selected output parameter.

Parameter Properties

The following table describes the properties for input and output parameters:

Property	Parameter Type	Description
CanUpdate	Input	Select this check box to indicate that this parameter can be updated if it is used in a data binding operation.
DataType	Input/Output	Specify the data type of the parameter. For more information, see Data Types.
Description	Input/Output	Specifies more information that a user may find helpful.
Display Name	Input/Output	Specifies the name to display in the Workflow Editor .
InvalidText	Output	Specifies text that will not validate when entered in the text box. This property is not available for Silverlight forms.

Property	Parameter Type	Description
IsRequired	Output	<p>Select this check box to indicate that the selected output parameter is required for data binding.</p> <p>This property is not available for Silverlight forms.</p>
IsValid	Output	<p>This check box is selected when the value is valid.</p> <p>This property is not available for Silverlight forms.</p>
Localization	Input/Output	<p>Click the ellipsis button to open the Localization Configuration dialog box where you can configure the property to localize, the language to use for localization, and the text to localize.</p> <p>This property is not available for Silverlight forms.</p>
Name	Input/Output	<p>Specifies the identifying name. The name provides a reference so that code behind, such as event handler code, can refer to a markup element during processing by a XAML processor.</p> <p>This property is not available for Silverlight forms.</p>
Uid	Input/Output	<p>Specifies the unique identifier (for localization).</p> <p>This property is not available for Silverlight forms.</p>
Value	Input/Output	<p>Specifies the current value of the parameter.</p>

Property	Parameter Type	Description
		 Important: Literal values for output parameters are not supported on Silverlight forms.

Add parameters to a form

Procedure

1. In the navigator, click **Global Displays**.
2. In the **Folders** list, select the resource type you want to work with.



Tip:

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click the **Parameters** tab.
5. To add an input parameter, in the Input Parameters area, click **Add**.
The input parameter displays in the list.
6. To add an output parameter, in the Output Parameters area, click **Add**.
The output parameter displays in the list.
7. Select the parameter, and then click the **Properties** tab.
8. Configure the properties, as required.



Note:

To view the full list of properties, select the **Advanced** check box.

9. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Remove parameters from a form

About this task



CAUTION:

If you delete or rename a control, parameter, or server method that is used as the source of a binding in a form, the binding itself is not removed from the XAML file. You can leave the binding code in the XAML, but if you create another control, parameter, or server method with the same name as the deleted object, or rename an existing object to make the object name valid again, then the orphaned binding may become valid as well, causing unexpected results.

It is recommended that you manually delete the binding code from the XAML file to prevent this issue from occurring.

Procedure

1. In the navigator, click **Global Displays**.
2. In the **Folders** list, select the resource type you want to work with.



Tip:

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click the **Parameters** tab.
5. To remove an input parameter, select the parameter you want to remove, and then in the Input Parameters area, click **Remove**.
The input parameter is removed from the list.
6. To remove an output parameter, select the parameter you want to remove, and then in the Output Parameters area, click **Remove**.
The output parameter is removed from the list.

7. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Designer: Data Items Panel

The Data Items panel allows you to configure data items that you can use to bind control properties to.

The following table describes the attributes of this panel:

Attribute	Value	Description
Data Items	User-defined	Displays the list of data items.
Data Items: Add	N/A	Click to add a data item.
Data Items: Remove	N/A	Click to remove a data item.

Data Item Properties

The following table describes the properties for data items. They should be configured at design time.

Property	Property Type	Description
Description	Input	Specifies more information that a user may find helpful.
DisplayName	Input	Specifies the name to display in the Workflow Editor .
Localization	Input	Click the ellipsis button to open the Localization Configuration dialog box where you can configure the property to localize, the language to use for localization, and the text to localize.
Name	Input	Specifies the identifying name. The name provides a reference so that code behind, such as

Property	Property Type	Description
		event handler code, can refer to a markup element during processing by a XAML processor.
Source	Input	<p>Click the ellipsis button to open the Configure a Data Source dialog box to select and configure a source for the data item. For more information, see Configure a data source item for a resource source.</p> <p>If you are developing a form and choose not to specify a data source, you can specify one in the workflow and pass it in.</p> <div data-bbox="1026 961 1421 1499" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: If you specify a data source in the form, it acts as a default value. Therefore, if you specify a value in the form and as input in the workflow, the value in the workflow overrides the value in the form, and is the value used at run time.</p> </div>
Source Address	Input	Specifies the fully qualified address of the data source you want to assign to the current data item. It can only be used for items with the same configuration. For example, this property can be used on all Mixer classes that have a Speed property.

Property	Property Type	Description
		<div data-bbox="1024 268 1421 1064" style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px;">  Note: Use this property only if you want to override the Ldap address of the item to read what is passed in by the workflow, but retain all other configuration details. For example, if you configure a workflow to read data items of Line1 (configured to read property-XYZ), you can override the source address to be Line2, and still read propertyXYZ. </div>
Subscribe	Input	<p>Select this check box to indicate that this data item should be subscribed to.</p> <p>If this check box is cleared, the data item value is read only once when the form or user display loads. When this check box is selected, the data item value is updated continuously as the data item source updates.</p>
Uid	Input	Specifies the unique identifier (for localization).

Add a data item to a form or user display

Procedure

1. In the navigator, click **Global Displays**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click the **Designer** tab.
5. Click the **Data Items** tab.
6. Click **Add**.
A new data item displays in the list.
7. Select the data item, and then click the **Properties** tab.
8. Configure the properties, as required.

**Note:**

To view the full list of properties, select the **Advanced** check box.

9. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Remove a data item from a form or user display

Procedure

1. In the navigator, click **Global Displays**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

- In the **Displays** panel, select an editor.

The editor appears in the workspace, displaying the resource information.

- Click the **Designer** tab.

- Click the **Data Items** tab.

- Select the data item you want to remove, and then click **Remove**.

The data item is removed from the list.

- Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Designer: Form Events Panel

The Form Events panel allows you to configure events that can be raised from this form. The form events you configure are available to workflows using this form.

The following table describes the attributes of this panel:

Attribute	Value	Description
Form Events	User-defined	Displays the list of events.
Form Events: Add	N/A	Click to add an event.
Form Events: Remove	N/A	Click to remove an event.

Form Event Properties

The following table describes the properties for form events:

Property	Property Type	Description
Description	Input	Specifies more information that a user may find helpful.
DisplayName	Input	Specifies the name to display in the Workflow Editor .
EventTrigger	Input	Click the ellipsis button to open the Event Triggers dialog box

Property	Property Type	Description
		where you can select the event(s) that will cause this method to be executed.
IsEnabled	Input	Select this check box to indicate that this event is enabled.
Localization	Input	Click the ellipsis button to open the Localization Configuration dialog box where you can configure the property to localize, the language to use for localization, and the text to localize.
Name	Input	Specifies the identifying name. The name provides a reference so that code behind, such as event handler code, can refer to a markup element during processing by a XAML processor.
Uid	Input	Specifies the unique identifier (for localization).

Related reference

[Event Triggers \(on page 1115\)](#)

Add a form event

Procedure

1. In the navigator, click **Global Displays**.
2. In the **Folders** list, select the resource type you want to work with.



Tip:

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.

The editor appears in the workspace, displaying the resource information.

4. Click the **Designer** tab.
5. Click the **Forms Events** tab.
6. Click **Add**.

The form event displays in the list.

7. Select the form event.
8. Configure the properties, as required.



Note:

To view the full list of properties, select the **Advanced** check box.

9. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Remove a form event

Procedure

1. In the navigator, click **Global Displays**.
2. In the **Folders** list, select the resource type you want to work with.



Tip:

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click the **Designer** tab.
5. Click the **Forms Events** tab.
6. Select the event you want to remove, and then click **Remove**.
The form event is removed from the list.
7. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Event Triggers

The **Event Triggers** dialog box allows you to select the event you want to raise when an action is performed on a control in a form.

The **Event Triggers** dialog box displays the applicable events for each of the configuration types you have defined. For example, if you configured parameters and server methods for a form, the events applicable to those types are displayed and grouped according to type.

The following table describes the events that can be configured.

Event	Occurs when...	Applies to...
Checked	a control is checked	<ul style="list-style-type: none"> • Check Box • Radio Button • Toggle Button • Menu Item
Click	a control is clicked	<ul style="list-style-type: none"> • Button • Check Box • Radio Button • Repeat Button • Toggle Button • Menu Item • Switch Language Button
Collapsed	the content window of a control closes and only the header is visible	<ul style="list-style-type: none"> • Expander • Tree View Item
Date Time Changed	a date time is selected	Date/Time Picker
Drag Completed	the Thumb control loses mouse focus	Grid Splitter

Event	Occurs when...	Applies to...
Drag Delta	the mouse changes position when a Thumb control has logical focus and mouse capture	Grid Splitter
Drag Started	a Thumb control receives logical focus and mouse capture	Grid Splitter
Drop Down Closed	the drop-down list of the combo box closes	Combo Box
Drop Down Opened	the drop-down list of the combo box opens	Combo Box
Duration Changed	a duration is selected	Duration Picker
ExecutedSuccessfully	the method executed successfully	<ul style="list-style-type: none"> • Server methods • Client methods
ExecutedWithError	the method fails	<ul style="list-style-type: none"> • Server methods • Client methods
Expanded	the content window of a control opens to display both its header and content	<ul style="list-style-type: none"> • Expander • Tree View Item
Got Focus	the control gets focus	All controls
Indeterminate	the state of a control is neither on nor off	<ul style="list-style-type: none"> • Check Box • Radio Button • Toggle Button
Key Down	a key is pressed while focus is on the control	All controls
Layout Updated	the layout of the various visual elements associated with the current Dispatcher changes	All controls
Loaded	the control is laid out, rendered, and ready for interaction	All controls
Lost Focus	the control loses focus	All controls

Event	Occurs when...	Applies to...
Mouse Double Click	a mouse button is clicked two or more times	All controls
Mouse Down	any mouse button is pressed while the pointer is over the control	All controls
Mouse Enter	the mouse pointer enters the bounds of the control	All controls
Mouse Leave	the mouse pointer leaves the bounds of the control	All controls
Mouse Left Button Down	the left mouse button is pressed while the mouse pointer is over the control	All controls
Mouse Move	the mouse pointer moves while over the control	All controls
Mouse Right Button Down	the right mouse button is pressed while the mouse pointer is over the control	All controls
Page View Changed	the collection of Document-PageView items associated with this viewer (represented by the PageViews property) changes	Document Viewer
Password Changed	the value of the Password property changes	Password Box
Scroll	the scroll bar moves	Scroll Bar
Scroll Changed	changes are detected to the scroll position, extent, or viewport size	Scroll Viewer
Selected	the IsSelected property of changes from false to True	<ul style="list-style-type: none"> • Combo Box Item • List Box Item • List View Item • Tree View Item

Event	Occurs when...	Applies to...
Selection Changed	the text selection has changed	<ul style="list-style-type: none"> • Rich Text Box • Text Box • Validating Text Box • Combo Box • List Box • List View • Tab Control • Universal Browser
Selection Item Changed	the selected item has changed	Tree View
Submenu Opened	the state of the IsSubmenuOpen property changes to True	Menu Item
Submenu Closed	the state of the IsSubmenuOpen property changes to false	Menu Item
Text Changed	content changes in the text element	<ul style="list-style-type: none"> • Masked Text Box • Rich Text Box • Text Box • Validating Text Box
Unchecked	a control is unchecked	<ul style="list-style-type: none"> • Check Box • Radio Button • Toggle Button • Menu Item
Unloaded	the element is removed from within an element tree of loaded controls	All controls
Unselected	the IsSelected property changes from true to False	<ul style="list-style-type: none"> • Combo Box Item • List Box Item • List View Item • Tree View Item
Value Changed	the value changes	<ul style="list-style-type: none"> • Progress Bar • Scroll Bar • Slider • Input Parameter

Event	Occurs when...	Applies to...
		<ul style="list-style-type: none"> • Output Parameter • Data Item

Designer: Server Methods

The Server Methods panel allows you to configure service interfaces that can be called from the current form or user display. The server methods available are also used to configure call methods in workflows.

Each method exposes two events: **ExecutedSuccessfully** and **ExecutedWithError**. For more information, see *Server and Client Methods Events*.

The following table describes the attributes of this panel:

Attribute	Value	Description
Server Methods	User-defined	Displays the list of methods.
Server Methods: Add	N/A	Click to add a method.
Server Methods: Remove	N/A	Click to remove a method.

Server Method Properties

The following table describes the properties for server methods:

Property	Property Type	Description
DisplayName	Input	Specifies the name to display in the Workflow Editor .
EventTrigger	Input	Click the ellipsis button to open the Event Triggers dialog box where you can select the event(s) that will cause this method to be executed.
IsEnabled	Input	Select this check box to indicate that this event is enabled. By default, this check box is selected. However, you can bind its value to an expression so that

Property	Property Type	Description
		the method is only conditionally executed. For example, you can bind the IsEnabled property of the method to the IsChecked property of a check box so that the method never executes if the check box is not selected.
Localization	Input	Click the ellipsis button to open the Localization Configuration dialog box where you can configure the property to localize, the language to use for localization, and the text to localize.
Name	Input	Specifies the identifying name. The name provides a reference so that code behind, such as event handler code, can refer to a markup element during processing by a XAML processor.
Source	Input	Click the ellipsis button to open the Configure Method dialog box where you can select the method to use.
Uid	Input	Specifies the unique identifier (for localization).

**Note:**

When you select a method from the Configure Method dialog box, the input properties specific to that method are displayed in the Properties tab. For more information on the specific method properties, see *Service Interfaces*.

**Tip:**

To view the method-specific properties in a separate group, click the **By Category** button to sort the properties.

Related reference

[Server and Client Methods Events \(on page 1147\)](#)

Add a server method to a display

Procedure

1. In the navigator, click **Global Displays**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, click **Display Editor**.
4. Click the **Server Methods** tab.
5. Click **Add**.
A server method displays in the list.
6. Select the server method.
7. Configure the properties, as required.

**Note:**

To view the full list of properties, select the **Advanced** check box.

8. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Remove a server method from a display

About this task**CAUTION:**

If you delete or rename a control, parameter, or server method that is used as the source of a binding in a form, the binding itself is not removed from the XAML file. You can leave the binding code in the XAML, but if you create another control, parameter, or server method with the same name as the deleted object, or rename an existing object to make the object name valid again, then the orphaned binding may become valid as well, causing unexpected results.

It is recommended that you manually delete the binding code from the XAML file to prevent this issue from occurring.

Procedure

1. In the navigator, click **Global Displays**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, click **Display Editor**.
4. Click the **Server Methods** tab.
5. Select the server methods you want to remove, and then click **Remove**.
The server method is removed from the list.
6. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Bind a return value to a DataGrid using form server methods

To display a data table in a Silverlight form, you must bind the return value using server methods.

Before you begin

Create a Silverlight form for web-based content.

About this task



Note:

The information in this topic is not related to WPF forms unless otherwise stated.

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. Click **Display Editor > Silverlight Config**.
4. Click **Add**.
Your **Windows** browser appears.
5. From the <install dir>\Microsoft SDKs\Silverlight\v5.0\Libraries\Client folder, select `System.Windows.Controls.Data.dll`, and then click **Open**.
6. **Optional:** In the **Description** field, enter new information about the file, and then click **Save**.
7. Click the **Designer** tab.
8. Click the **Server Methods** tab.
9. Click **Add**.
10. In the **Properties** panel, in the **Source** field, click the ellipsis button.
The **Pick a method** dialog box appears.
11. Expand **IDatabaseService**, select **ExecuteStatement**, and then click **Next**.
12. In the **Statement** section, click **Browse** next to the **Use Existing** radio button.
The Universal Browser appears.
13. Select a statement, click **OK**, and then click **Finish**.
14. In the **Connection** field, click the ellipsis button.
The Universal Browser appears.
15. Expand **Databases > Database Connection**, select a connection, and then click **OK**.
16. From the **EventTriggers** field, click the **Add** button.
The **Event Triggers** dialog box appears.
17. Select the event that you want to trigger the method.
18. Click **Save**.
19. Click the **Project** tab.
20. Expand <YOURPROJECTNAME> > **MainPage.xaml**.
21. Double-click **MainPage.xaml.cs**.
22. Add the code in bold to this page in the location indicated. Edit column names and add additional columns as appropriate for your call.

```
namespace YOURPROJECTNAME
{
```

```

public partial class MainPage : UserControl, IFragment
{
    [Import]
    public IFeedbackService FeedbackService { get; set; }

    readonly ILog _logger = LogManagerEx.GetLogger("YOURPROJECTNAME");
    readonly Dictionary<Type, object> _interfaceHelpers = new Dictionary<Type, object>();

    public MainPage()
    {
        ProficiencyComposition.SatisfyImports(this);
        InitializeComponent();

        _interfaceHelpers[typeof(IResourceParameters)] = new FragmentMetadataByAttributes(this);
        _interfaceHelpers[typeof(IFragmentInitialization)] = new FragmentInitialization(this);

        InitializeProficiencyData();

        DataGrid1.AutoGenerateColumns = false;

        DataGrid1.Columns.Add(new DataGridTextColumn() { Header = "Header1", Binding = new
Binding("[YOURCOLUMNNAME]") });
        DataGrid1.Columns.Add(new DataGridTextColumn() { Header = "Header2", Binding = new
Binding("[YOURCOLUMNNAME2]") });

    }

    public T GetInterface<T>() where T : class
    {
        if (_interfaceHelpers.ContainsKey(typeof(T)))
        {
            var iface = _interfaceHelpers[typeof(T)] as T;

            Debug.Assert(iface != null, "_interfaceHelpers[" + typeof(T).Name + "] didn't have an object of the right
type.");

            return iface;
        }

        return this as T;
    }
}

```

```
}
}
```

23. Click **Save**.
24. Click the **MainForm.xaml** tab.
25. Click the **Toolbox** tab.
26. Expand **Custom Controls**, and then select **DataGrid**.
 - For a WPF form, expand **Grids and Panels**.
27. Drag the DataGrid control onto `MainPage.xaml`.
28. From the **Properties** panel, go to the **ItemsSource** property, click the drop-down arrow, and then select **Bind**.

The **Binding** dialog box appears.
29. Select **Server Methods** > <YOURMETHODNAME> > **ReturnValue** > **FirstTable** > **Rows**.
 - For a WPF form, bind the **InputTable** property of the Data Grid control to the result of the server method.
30. Click **Save**.

Use a service provider method exclusively in code

Most service provider methods are accessible and can be configured using **Service Provider Methods**. For those that are not accessible, follow the tips in this topic for assistance.



Important:

The following procedure requires the adding of additional assembly references to the form. Any additional references should be added through the **Designer** to ensure that they are added properly. If a reference is added in Visual Studio, ensure that the `CopyLocal` value = `false` and the `SpecificVersion` value = `false` or the form may not be useable with future product releases.

In most cases, the required Service Provider methods, including those from custom providers, are accessible and can be configured through the Server Methods dialog. However, if a method is configured with the `Browsable` attribute set to `False`, it does not appear in the Server Methods dialog, and it can only be accessed through code.

In Designer-generated forms or user displays, access the `IServiceDirectory` service from the host as follows:

```
IServiceDirectory serviceDirectory = this.Host.GetService(typeof(IServiceDirectory)) as
IServiceDirectory;
```

You must have an assembly reference to

`Proficy.Platform.Core.ClientTools.ServiceDirectory.Interfaces.IServiceDirectory.dll` and the assembly containing the Service Provider interface you want to access. Additionally, have a using statement for `Proficy.Platform.Core.ClientTools.ServiceDirectory.Interfaces` and the Service Provider interface. The `IServiceDirectory` service provides access to Service Providers through their interface names as follows:

```
ICustomSP someCustomSP = serviceDirectory.CreateLogicalServiceProxy(typeof(ICustomSP).FullName)
as ICustomSP;
```

For example, an `IEquipment` service provider method can be executed with the following code:

```
IEquipment equipmentSP = serviceDirectory.CreateLogicalServiceProxy(typeof(ICustomSP).FullName)
as ICustomSP; Proficy.Platform.Forms.Editor.CustomControls.UniversalBrowser browser =
UniversalBrowser1; int result = equipmentSP.QueryPropertyCount(browser.SelectedItem as
DirectoryResource,"%"); TextBox tb = TextBox1; tb.Text= result.ToString();
```

For this code to work, add an assembly reference to `Proficy.Platform.Services.EquipmentInterfaces.dll`. Also add the following using statements:

- using `Proficy.Platform.Services.Equipment.Interfaces`
- using `Proficy.Platform.Core.ProficySystem.Types`

For a custom service provider, add the appropriate assembly and namespace references.

Designer: Client Methods

The **Client Methods** panel allows you to configure client-specific methods that can be called from the current form or user display.

The available client methods are specific to actions that can take place within Workflow, within a form or user display, or within a form in a running workflow.

Each method exposes two events: **ExecutedSuccessfully** and **ExecutedWithError**. For more information, see *Server and Client Methods Events*.

The following table describes the attributes of this panel:

Attribute	Value	Description
Client Methods	User-defined	Displays the list of methods.
Client Methods: Add	N/A	Click to add a method.

Attribute	Value	Description
Client Methods: Remove	N/A	Click to remove a method.

Client Method Properties

The following table describes the properties for client methods:

Property	Property Type	Description
DisplayName	Input	Specifies the name to display in the Workflow Editor .
EventTrigger	Input	Click the ellipsis button to open the Event Triggers dialog box where you can select the event(s) that will cause this method to be executed.
IsEnabled	Input	Select this check box to indicate that this event is enabled. By default, this check box is selected. However, you can bind its value to an expression so that the method is only conditionally executed. For example, you can bind the IsEnabled property of the method to the IsChecked property of a check box so that the method never executes if the check box is not selected.
Localization	Input	Click the ellipsis button to open the Localization Configuration dialog box where you can configure the property to localize, the language to use for localization, and the text to localize.
Name	Input	Specifies the identifying name. The name provides a reference

Property	Property Type	Description
		so that code behind, such as event handler code, can refer to a markup element during processing by a XAML processor.
Source	Input	Click the ellipsis button to open the Configure Method dialog box where you can select the method to use.
Uid	Input	Specifies the unique identifier (for localization).

**Tip:**

To view the method-specific properties in a separate group, click the **By Category** button to sort the properties.

Related reference

[Server and Client Methods Events \(on page 1147\)](#)

Client Methods

You can configure client methods to be called from the WPF Designer.

**Note:**

Client methods are available only in the WPF form editor, and work only on the Task List and Mobile-sized Task List (not the Web Task List).

Source Methods

After you add a client method, you select one of the following source methods to execute. You then configure an event trigger to determine when these methods should execute.

Related reference

[Event Triggers \(on page 1115\)](#)

Add a client method to a form or display

Prerequisites

[Client Methods \(on page 1128\)](#)

Procedure

1. In the navigator, click **Global Displays**.
2. In the **Folders** list:

To work with a...	Select the resource you want to work with from the...
Display	Displays list
Form	Forms list

3. In the **Displays** panel, click **Display Editor**.
4. Click the **Designer** tab.
5. Click the **Client Methods** tab.
6. Click **Add**.
A client method appears in the list.
7. Select the client method, and then click the **Properties** tab to configure the method.
8. Configure the properties, as required.



Note:

To view the full list of properties, select the **Advanced** check box.

9. In the **Source** field, click the ellipsis button.
The **Configure Method** dialog box appears.
10. Select the appropriate client method that you want to configure, and then click **Finish**.
11. Based on the source selected, configure any further resources, if required.
12. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Related reference

[Designer: Client Methods \(on page 1126\)](#)

Delete a client method from a form or display

Procedure

1. In the navigator, click **Global Displays**.
2. In the **Folders** list:

To work with a...	Select the resource you want to work with from the...
Display	Displays list
Form	Forms list

3. In the **Displays** panel, click **Display Editor**.
4. Click the **Designer** tab.
5. Click the **Client Methods** tab.
6. Select the client method you want to remove, and then click **Remove**.
The client method is removed from the list.
7. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Client Context

You can configure client methods to be called from the WPF Designer.

Attribute	Description
ChangeCurrentDisplay	Changes the current user display. Use the Display property to specify the display.

Attribute	Description	
ChangeNavigatorSelection		Changes the currently selected resource in the navigator. Use the Resource property to specify the resource.
GetEquipmentContext		Shows the equipment context used in a workflow Task List. Returns the equipment context in use in the workflow Task List.
SetEquipmentContext		Sets the equipment context in a workflow Task List. Use the Equipment property to specify the equipment context in order to filter the Task List.
GetLoggedInUser		Returns the user that is currently logged in.
GetNavigatorSelection		Returns the item selected in the navigator.
SubscribeToNavigatorSelection		<p>Subscribes to the navigator context and returns the current navigator selection through the ReturnValue of this method. As the navigator selection changes, the ReturnValue updates automatically.</p> <p>This object also exposes a NavigatorSelectionChanged event that can be used as the EventTrigger for a server method or from code-behind .</p> <div data-bbox="1024 1717 1419 1869" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Tip: Use the EventTrigger to specify when to</p> </div>

Attribute	Description	
		 start subscribing to the navigation selection changes, which is usually on the loaded event of your display.
VerifyUserCredentials		Verifies a user's credentials during run time.

Related reference

[Designer: Server Methods \(on page 1119\)](#)

Configure linked equipment for forms

You can use client methods to view and change pieces of equipment during run time.

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. Add a form, or from the **Forms** list, select an existing form.
3. In the **Displays** panel, click **Display Editor > Designer**.
4. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
5. Click the **Client Methods** tab.
6. Click **Add**.
A client method appears in the list.
7. In the **Source** field, click the ellipsis button.
The **Configure Method** dialog box appears.
8. To set the equipment context, in the **Client Context** section, select **SetEquipmentContext**, and then click **Finish**.
9. In the **Equipment** field:

Click the...	To...
ellipsis button	add a piece of equipment from the Universal Browser.

Click the...	To...
drop-down arrow	bind the equipment to a directory resource variable in order for a workflow to select a piece of equipment at run time.

**Note:**

You cannot use Equipment Classes for equipment changes.

10. **Optional:** To get the equipment context, in the **Client Context** section, select **GetEquipmentContext**, and then click **Finish**.
11. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Forms

Overview

Forms are displays that are used as activities in workflows, providing a user interface for task steps. A form can display data or allow you to interact with a running process, or both.

You can significantly improve the performance of your workflows by implementing one or all of the following recommendations:

- Reduce Form Parameters
- Bind in the Form Control Property
- Change the Object Property
- Bind on Controls

Client Controls

You can configure client methods to be called from the WPF Designer.

Attribute	Description	
GetItemFromItemsControl		Use the Index property to get an item from the Items Control group (such as, ListBox, ComboBox, and so on). Use the Control property to specify the Items Control.
SelectInItemsControl		Use the Index property to select an item from the Items Control group (such as, ListBox, ComboBox, and so on). Use the Control property to specify the Items Control.
SetFocus		Sets a control's focus to a specified control. Use the Control property to specify the control to set focus to.

Diagnostics

You can configure client methods to be called from the WPF Designer.

Attribute	Description	
ClearFeedbackZone		Clears messages written from the WriteToFeedbackZone client method.
WriteToClientLog		Use the Message property to write a message to a log file.
WriteToFeedbackZone		Use the Message property to write a message to the Feedback Zone. You can also use the Log property to specify whether this message should be logged to the Client log file.

Forms

You can configure client methods to be called from the WPF Designer.

The following methods are available when configuring forms only.

Attribute	Description	
AcquireTaskStep		Allows a form to acquire the selected task step.
CancelForm		Cancels the current form.
ReleaseTaskStep		Allows a form to release the selected task step.
Set Selected Task		When set, the task, based on the task name, immediately selects the task if it exists in the list. If the task is not in the list, it caches the task name and searches for it the next time it is looking for a task to run.
Set Selected Task Step		When set, the task step, based on the task step name, immediately selects the task step if it exists in the list.
SubmitForm		Submits the current form.

Acquire or release task steps in forms

Before you begin

You can include acquire and release buttons on any form you create.

Procedure

1. In the navigator, click **Global Displays**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

- In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
- Click the **Client Methods** tab.
- Click **Add**.
A client method appears in the list.
- In the **Source** field, click the ellipsis button.
The **Configure Method** dialog box appears.
- In the **Forms** section, select one of the following options, and then click **Finish**:

Click...	To allow a workflow to...
AcquireTaskStep	acquire a task step.
ReleaseTaskStep	release a task step.

- Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Related reference

[Client Methods \(on page 1128\)](#)

Select a task automatically within a form during workflow execution using client methods

You can set a task or task step to be selected automatically in the Task List during run time by configuring client methods with the Set Selected Task feature.

Procedure

- From the navigator, click **Global Displays > Forms**.
- Add a form, or from the **Forms** list, select an existing form.

3. Click **Display Editor > Designer**.
4. Click the **Client Methods** tab.
5. Click **Add**.
A client method appears in the list.
6. Select the client method, and then click the **Properties** tab to configure the method.
7. In the **Source** field, click the ellipsis button.
The **Configure Method** dialog box appears.
8. Configure the properties, as required.

**Note:**

To view the full list of properties, select the **Advanced** check box.

9. Select one of the following client methods, and then click **Finish**.
 - To automatically select a task at the workflow level, click **Set Selected Task**.
 - To automatically select a task step at the subprocess level, click **Set Selected Task Step**.
10. Based on the source selected, configure the following property.
 - If you set a task, in the **Task Name** field, enter the related name.
 - If you set a task step, in the **Task Step Name** field, enter the related name.
11. If required, create an expression or perform a bind to set parameter values.
12. Click **Save**.

Task List Filters

You can configure client methods to be called from the WPF Designer.

You can add a filter to a task property so that, during run time, the Task List is updated using these filters. Filtering in this way makes a specific task readily available for an operator when it is time to perform that task.

**Note:**

These filters are unavailable when using a web-based version of the Task List.

Attribute	Description
Clear Filter	Clears a specified previously set filter. After selecting this method source, the Filter Type field is available to select which avail-

Attribute	Description	
		able Task List filter you want to clear, as well as perform a bind.
Set Categories Filter		Filters the Task List based on the workflow categories assigned to tasks or task steps. After selecting this method source, the Cate- gories field is available to set the resources, as well as perform a bind.
Set Expires Before Filter		Sets a filter for a date and time that occur before a task expires. After selecting this method source, the Expiry field is available to set the date and time, as well as perform a bind or expression.
Set Expires In Filter		Sets a filter for a time that occurs within a set timeframe before a task expires. After selecting this method source, the Expiry field is available to set the time, as well as perform a bind or expression.
Set Location Assignment Filter		Filters the Task List based on the location or equipment that a task is assigned to. After selecting this method source, the Loca- tion Assignment field is available to set the resource(s), as well as perform a bind.
Set My Task Steps Filter		Filters the Task List based on the task steps for operator who is logged in. After selecting this method source, the Filter My Task Steps field is available to

Attribute	Description	
		select either True or False, as well as perform a bind or expression.
Set Name Filter		Filters the Task List based on the name given to a task. After selecting this method source, the Task Name Filter field is available to insert a string containing the name you want to sort the Task List by. You can also perform a bind or expression based on this value.
Set Personnel Assignment Filter		Filters the Task List based on the person that a task is assigned to. After selecting this method source, the Personnel Assignment field is available to set the resources, as well as perform a bind.
Set Predefined Filters		Applies the selected predefined filters to the Task List, clearing any previously applied filters. Filters the Task List based on these filters.
Set Priority Filter		Filters the Task List based on the priority number given to a task. After selecting this method source, the Priority Filter field is available to insert a number that corresponds to the priority level of the task. You can also perform a bind or expression based on this value.

Clear a Task List filter setting

Procedure

1. From the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. In the **Displays** panel, click **Global Information > Display Editor**.
4. On the **Overview** tab, select **Use the Designer tab to create new rich client content**.
5. Click the **Designer** tab.
6. Click the **Client Methods** tab.
7. Click **Add**.
A client method appears in the list.
8. Select the method, and then click the **Properties** tab.
9. In the **Source** field, click the ellipsis button.
The **Configure Method** dialog box appears.
10. In the **Task List Filters** section, select **Clear Filter**, and then click **Finish**.
11. In the **Filter Type** field, from the drop-down menu, select a previously set filter that you want to clear.
12. If applicable, to perform a bind or reset, click the drop-down arrow.
13. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Set a Task List expiry filter

Procedure

1. From the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. In the **Displays** panel, click **Global Information > Display Editor**.
4. On the **Overview** tab, select **Use the Designer tab to create new rich client content**.
5. Click the **Designer** tab.
6. Click the **Client Methods** tab.
7. Click **Add**.
A client method appears in the list.
8. Select the method, and then click the **Properties** tab.

9. In the **Source** field, click the ellipsis button.
The **Configure Method** dialog box appears.
10. In the **Task List Filters** section, select one of the following options, and then click **Finish**.

Click...	To select a filter for...
Set Expires Before Filter	tasks that expire before a given date/time.
Set Expires In Filter	tasks that expire within a given time span.

11. In the **Expiry** field, use the arrows to set the date and/or time.
12. If required, to perform a bind or reset, click the drop-down arrow.
13. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Set a Task List filter based on location

You can filter the Task List by the location or piece of equipment that a task is assigned to.

Procedure

1. From the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. In the **Displays** panel, click **Global Information > Display Editor**.
4. On the **Overview** tab, select **Use the Designer tab to create new rich client content**.
5. Click the **Designer** tab.
6. Click the **Client Methods** tab.
7. Click **Add**.
A client method appears in the list.
8. Select the method, and then click the **Properties** tab.
9. In the **Source** field, click the ellipsis button.
The **Configure Method** dialog box appears.
10. In the **Task List Filters** section, select **Set Location Assignment Filter**, and then click **Finish**.
11. In the **Location Assignment** field, click the ellipsis button.
The **Universal Browser** appears.
12. Navigate to and select the resources, then click **OK**.

**Tip:**

You can select several resources at the same time.

13. If applicable, to perform a bind or reset, click the drop-down arrow.
14. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Set a Task List filter based on personnel

You can filter the Task List by the person that a task is assigned to.

Procedure

1. From the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. In the **Displays** panel, click **Global Information > Display Editor**.
4. On the **Overview** tab, select **Use the Designer tab to create new rich client content**.
5. Click the **Designer** tab.
6. Click the **Client Methods** tab.
7. Click **Add**.
A client method appears in the list.
8. Select the method, and then click the **Properties** tab.
9. In the **Source** field, click the ellipsis button.
The **Configure Method** dialog box appears.
10. In the **Task List Filters** section, select **Set Personnel Assignment Filter**, and then click **Finish**.
11. In the **Personnel Assignment** field, click the ellipsis button.
The **Universal Browser** appears.
12. Navigate to and select the resources, then click **OK**.

**Tip:**

You can select several resources at the same time.

13. If applicable, to perform a bind or reset, click the drop-down arrow.

14. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Filter an operator's task steps

Procedure

1. From the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. In the **Displays** panel, click **Global Information > Display Editor**.
4. On the **Overview** tab, select **Use the Designer tab to create new rich client content**.
5. Click the **Designer** tab.
6. Click the **Client Methods** tab.
7. Click **Add**.

A client method appears in the list.

8. Select the method, and then click the **Properties** tab.
9. In the **Source** field, click the ellipsis button.
The **Configure Method** dialog box appears.
10. In the **Task List Filters** section, select **Set My Task Steps Filter**, and then click **Finish**.
11. In the **Filter My Task Steps** field, from the drop-down menu, select one of the following.

Click...	To...
True	filter by an operator's task steps.
False	leave task steps unfiltered.

12. If applicable, to perform a bind or reset, click the drop-down arrow.
13. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Filter by a task's name

Procedure

1. From the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. In the **Displays** panel, click **Global Information > Display Editor**.
4. On the **Overview** tab, select **Use the Designer tab to create new rich client content**.
5. Click the **Designer** tab.
6. Click the **Client Methods** tab.
7. Click **Add**.
A client method appears in the list.
8. Select the method, and then click the **Properties** tab.
9. In the **Source** field, click the ellipsis button.
The **Configure Method** dialog box appears.
10. In the **Task List Filters** section, select **Set Name Filter**, and then click **Finish**.
11. In the **Task Name Filter** field, enter a string that references a workflow's task name, or part of a task name.
12. If applicable, to perform a bind or reset, click the drop-down arrow.
13. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Filter by a task's priority

Procedure

1. From the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. In the **Displays** panel, click **Global Information > Display Editor**.
4. On the **Overview** tab, select **Use the Designer tab to create new rich client content**.
5. Click the **Designer** tab.
6. Click the **Client Methods** tab.
7. Click **Add**.
A client method appears in the list.
8. Select the method, and then click the **Properties** tab.

9. In the **Source** field, click the ellipsis button.
The **Configure Method** dialog box appears.
10. In the **Task List Filters** section, select **Set Priority Filter**, and then click **Finish**.
11. In the **Priority** field, enter a number that relates to a previously set priority.
12. If applicable, to perform a bind or reset, click the drop-down arrow.
13. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Set a predefined filter

You can filter the Task List by predefined filters, which have been configured using the **Filter Editor**.

Procedure

1. From the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. In the **Displays** panel, click **Global Information > Display Editor**.
4. On the **Overview** tab, select **Use the Designer tab to create new rich client content**.
5. Click the **Designer** tab.
6. Click the **Client Methods** tab.
7. Click **Add**.
A client method appears in the list.
8. Select the method, and then click the **Properties** tab.
9. In the **Source** field, click the ellipsis button.
The **Configure Method** dialog box appears.
10. In the **Task List Filters** section, select **Set Predefined Filter**, and then click **Finish**.
11. In the **Predefined Filters** field, click the ellipsis button.
The **Universal Browser** appears.
12. Navigate to and select the resources, then click **OK**.



Tip:

You can select several resources at the same time.

13. If applicable, to perform a bind or reset, click the drop-down arrow.
14. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Filter by a category

You can filter the Task List by categories that have been preconfigured in a workflow.

Procedure

1. From the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. In the **Displays** panel, click **Global Information > Display Editor**.
4. On the **Overview** tab, select **Use the Designer tab to create new rich client content**.
5. Click the **Designer** tab.
6. Click the **Client Methods** tab.
7. Click **Add**.
A client method appears in the list.
8. Select the method, and then click the **Properties** tab.
9. In the **Source** field, click the ellipsis button.
The **Configure Method** dialog box appears.
10. In the **Task List Filters** section, select **Set Categories Filter**, and then click **Finish**.
11. In the **Categories** field, click the ellipsis button.
The **Universal Browser** appears.
12. Navigate to and select the resources, then click **OK**.



Tip:

You can select several resources at the same time.

13. If applicable, to perform a bind or reset, click the drop-down arrow.
14. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Server and Client Methods Events

Each method exposes event triggers that you can use to further configure your form or display.

Server and client methods expose the following event triggers: **ExecutedSuccessfully** and **ExecutedWithError**. In addition, if you select the **SubscribeToNavigatorSelection** source, the following event trigger is exposed: **NavigatorSelectionChanged**.

After a method executes, one of these events will occur, whether the method is triggered programmatically or via the events specified in the **EventTriggers** property.

ExecutedSuccessfully occurs if the method completes execution as expected. **ExecutedWithError** occurs if the method fails to complete as expected. For example, if you call the **CreateEquipment** method to create an Enterprise resource with an S95Id that is already in use, the **CreateEquipment** method returns an error stating that the specified name cannot be used and the **ExecutedWithError** event then occurs.

The **Event Triggers** dialog box can be accessed from either the **EventTriggers** property on the **Property** panel, or by right-clicking a control and selecting **Add Event Handler**. When you add an event handler, the control's code behind appears.

When adding an event handler for the **ExecutedWithError** event in code-behind, the exact exception causing the error is available as a parameter. For example, you can write the following code:

```
private void CreateEnterprisemethod_ExecutedWithError (object sender,
    Proficy.Platform.Core.DisplayFramework.Controls.ExecutionErrorEventArgs e)
{
    String errorMessageToUser = String.Format ("Oh no!! {0}", e.Exception.Message);
    MessageBox.Show(errorMessageToUser);
}
```

Additionally, when **ExecutedWithError** occurs, the property **LastExecutionError** on that method is set to the Exception object. This allows you to use this error without requiring any coding because it will be available in the Binding dialog box for use like any other property. If the method executes again, but this time successfully, **LastExecutionError** is reset to null. This property is read-only.

There are two scenarios for using these events. For each scenario, assume that you have already created two server methods with the Source property set to **CreateEquipment**: one you have set the S95 Type property to Enterprise and named it Create Enterprise, and the other you have set the S95 Type property to Site and named it Create Site.

Scenario A

You can use these events to raise form events. On the **Form Events** tab, add two form events: one called **ExecuteSuccess** to notify the workflow when the method executes successfully, and another called **ExecuteFailed** to notify the workflow when the method occurs with an error.

When you open the **Event Triggers** dialog box, the event options, **ExecutedSuccessfully** and **ExecutedWithError** are available under each of the server methods.

Scenario B

You can also use these events to chain methods together in the form. Using the Create Enterprise and Create Site methods, you can configure the Create Site method to create the site resource immediately after the enterprise resource, with the new enterprise as the parent. Configure the Create Site method as follows:

1. Select the `CreateEnterprise.ExecutedSuccessfully` event trigger.
2. Select `CreateEnterprise.ReturnValue` as the parent.

Programmatic Method Execution

A method can be executed programmatically using code-behind in conjunction with or instead of being triggered by events specified by the `EventTriggers` property.

Additionally, the `ExecutedSuccessfully` and `ExecutedWithError` events can be handled programmatically. For more information, see *Server and Client Method Events*.

All bindings to return values, as well as inputs, function the same whether the method is executed programmatically or as the result of a configured `EventTrigger`. To execute a method programmatically, use the following in the code-behind editor.

```
ExecuteMethod (string methodName)
```

For detailed steps, see *Execute a WPF Method Programmatically*.

Full example in VB for WPF

```
Imports System
Imports System.Collections
Imports System.Collections.Generic
Imports System.Data
Imports System.Windows
Imports System.Windows.Controls
Imports Proficy.Platform.Core.DisplayFramework.Forms
```

```

Partial Public Class MainForm

    Inherits FormBase

    Public Sub New()

        InitializeComponent()

        AddHandler Button1.Click, AddressOf Me.Button1_Click

    End Sub

    Private Sub Button1_Click(ByVal sender As Object, ByVal e As System.Windows.RoutedEventArgs)

        Me.ExecuteMethod("SampleMethod")

    End Sub

End Class

```

Full example in C# for WPF

```

using System;

using System.Collections;

using System.Collections.Generic;

using System.Data;

using System.Windows;

using System.Windows.Controls;

using Proficy.Platform.Core.DisplayFramework.Forms;

namespace Dev
{
    public partial class MainForm : FormBase
    {
        public MainForm()
        {
            InitializeComponent();

            Button1.Click += new System.Windows.RoutedEventHandler(this.Button1_Click);
        }
    }
}

```

```
private void Button1_Click(object sender, System.Windows.RoutedEventArgs e) {  
  
    this.ExecuteMethod("SampleMethod");  
  
}  
  
}  
  
}
```

Full example in C# for Silverlight

```
using System;  
  
using System.Collections;  
  
using System.Collections.Generic;  
  
using System.Data;  
  
using System.Windows;  
  
using System.Windows.Controls;  
  
using Proficy.Platform.Core.DisplayFramework.Forms;  
  
namespace Dev  
{  
    public partial class MainForm : FormBase  
    {  
        public MainForm()  
        {  
            InitializeComponent();  
  
            Button1.Click += new System.Windows.RoutedEventHandler(this.Button1_Click);  
        }  
  
        private void Button1_Click(object sender, System.Windows.RoutedEventArgs e) {  
            this.SampleMethod.Execute();  
        }  
  
    }  
}
```

Execute a WPF method programmatically

Procedure

1. From the navigator, click **Global Displays > Forms**.
2. Add a form, or from the **Forms** list, select an existing form.
3. Click **Display Editor > Designer**.
4. Click the **Method** tab.
5. Add a new method select it, and then click the **Properties** tab.
6. In the **Name** field, enter `SampleMethod`.
7. Click the **Parameters** tab, and then configure inputs, as required.
8. Click the **MainForm.xaml** tab.
9. Click the **Toolbox** tab.
10. Expand the **Buttons** control group, and then select a button.
11. Click and drag the control to the desired position in the **Designer**.



Note:

To apply the properties of a control to another control, click and drag the control to another control in the document tree. The Applying **<control (control name)> to Target** dialog box appears where you can select the option you want to use to apply to the control.

12. Right-click the button control and select **Add Event Handler**.
The **Event Triggers** dialog box appears.
13. Expand the button control, select **Click**, and then click **OK**.
The code-behind editor opens.
14. Add the call to **ExecuteMethod("SampleMethod")**.
15. Click **Save**.

Programmatic Parameter Inputs

Inputs can also be specified programmatically. This is achieved by accessing the method object by name, and then setting the value on the input, specified by either the name or the index.

Setting input by name:

```
C#: Method1.Inputs["S95Id"].Value = "enterprise123";
VB: Method1.Inputs("S95Id").Value = "enterprise123"
```

Setting input by index (C#). When specifying input by index, it is zero based. However, use caution with this mechanism because the index does not necessarily match the order in which the parameters are shown in the Properties panel. Therefore, this mechanism is useful only when there is one parameter.

```
C#: Method1.Inputs[0].Value = "enterprise123";  
VB: Method1.Inputs(0).Value = "enterprise123"
```

Parameters can also be iterated over in the following manner:

```
C#:  
  
    foreach (var parameter in Method1.Inputs)  
    {  
  
        String parameterName = parameter.ParameterName;  
  
        Object value = parameter.Value;  
  
    }  
  
VB:  
  
    Dim parameterName As String  
  
    Dim value As Object  
  
    For Each parameter in SampleMethod.Inputs  
  
        parameterName = parameter.ParameterName  
  
        value = parameter.Value  
  
    Next
```

Related reference

[Server and Client Methods Events \(on page 1147\)](#)

[Programmatic Method Execution \(on page 1148\)](#)

Designer: Test Panel

The **Test** panel allows you to view the results of a form or user display when you build and run in the Designer.

In order to view the results on the **Test** panel, you must configure values for the input parameters and data items. After the parameters and data items are configured, you can build and run the form or user display. The form or user display appears in a separate window and when the **Submit** button is pressed, the results appear in the **Event Protocol** area of the **Test** panel.



Note:

Parameters are available for forms only; they are not available for user displays.

View the results of a form or user display build

Before you begin

You must configure input parameters and data items for the form or user display in order to view the results in the Test tab.

**Note:**

Parameters are available for forms only; they are not available for user displays.

Procedure

1. In the navigator, click **Global Displays**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click the **Designer** tab.
5. On the toolbar, click **Build and Run**.
The Workflow Login dialog box appears.
6. Enter your username and password, and then click OK.
The form or user display builds, and then opens in a separate window.
7. Click **Submit**.
8. Click the **Test** tab.
The build results display in the Event Protocol area.
9. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Designer: Default Properties

The **Default Properties** panel allows you to configure default drawing properties for the control currently set as the active layer.

The following table describes the properties available in this dialog box:

Property	Description	
DecimalPlaces		Specifies the default number of decimal places used to display numbers.
Fill		Expand this field to select the default fill color for the active layer.
ShowGrid		Select this check box to indicate that grid lines are visible.
SnapToGrid		Select this check box to indicate that controls added to the active layer are snapped to the grid lines.
SnapToPixels		Select this check box to indicate that rendering for this control should use device-specific pixel settings.
Stroke		Expand this field to select the default color for the outline of the active layer.
StrokeThickness		Specifies the thickness, in pixels, of the stroke lines.

Related reference

[Designer Panels \(on page 710\)](#)

Related information

[Configure default properties for controls \(on page 1154\)](#)

Configure default properties for controls

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. In the **Folders** list, select the resource type you want to work with.

**Tip:**

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. Click the **Designer** tab.
5. From the toolbar, click **View Panels**, and then select **Default Properties**.
The **Default Properties** dialog box appears.
6. Select the control you want to configure.
7. Click the **Properties** tab.
8. Configure the properties, as required.

**Note:**

To view the full list of properties, select the **Advanced** check box.

9. Close the **Default Properties** dialog box.
10. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Designer: Preview Panel

The **Preview** panel allows you to view the form or user display before you build it.

**Note:**

The **Preview** panel is a read-only panel. You cannot make any changes to the form or user display.

Preview a form or user display

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.

3. Click **Display Editor > Designer**.
4. From the toolbar, click **View Panels**, and then select **Preview**.
The **Preview** dialog box appears.
5. View the form or user display and, if applicable, make note of any corrections you want to make to the appearance.

**Note:**

The Preview dialog box can be minimized while you make changes to the form or user display and is automatically updated with those changes.

6. Continue previewing the form or user display until you are satisfied with its appearance.
7. Close the **Preview** dialog box.
8. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Display and Group Duplication

Display and Group Duplication

Workflow allows you to duplicate displays and display groups.

In addition, you can specify where you want the copy to be located. That is, you can duplicate a form or user display and locate it in either the Displays or Forms folder as a new parent group, or you can locate it within an existing group as a child of that parent.

Duplicating displays is useful if you want to reuse certain aspects of a display without having to recreate it. Since you cannot move an existing display, the ability to duplicate a display under a different group means you do not have to manually recreate a display in order for it to belong to a group.

Duplication can also be used to move a form from one group to another. You can duplicate the form to a new location and then delete the original. However, if there are any workflows referencing the form in the original location, you must update each of the workflows that use that form to reference the new location.

**Note:**

When you duplicate a group, all of the displays that are part of that group are also duplicated.

Duplication works for all varieties of forms and displays, including those that are created in the Designer, uploaded as a .NET assembly, or ones that point to a web URL.

Related reference

[Form Design Environment \(on page 700\)](#)

[Basic Forms and Displays \(on page 1177\)](#)

Related information

[Form Templates \(on page 1160\)](#)

[Form Development in Visual Studio \(on page 1224\)](#)

Add displays or forms to a group

Procedure

1. In the navigator, click **Global Displays**.
2. In the **Folders** list, click **Displays** or **Forms**, and then from the appropriate **Groups** list, select the group you want to add a display or form to.
If you have nested groups, continue selecting the appropriate group in each list until you locate the group you want to add a display or form to.
3. Click **Add Display** or **Add Form**, and then add a display or form.
4. Click **OK**.

Related information

[Add a Global Displays resource \(on page 1181\)](#)

Add a display or form group to another group

Procedure

1. In the navigator, click **Global Displays**.
2. In the **Folders** list, click **Displays** or **Forms**, and then from the appropriate **Groups** list, select the group you want to add a display or form to.
If you have nested groups, continue selecting the appropriate group in each list until you locate the group you want to add a display or form to.
3. Click **Add Display** or **Add Form**, and then click **Add Display Group** or **Add Form Group**, as required.

4. In the **Name** and **Description** fields, enter a name and description for the resource.
5. Click **OK**.

Duplicate a display or group

This procedure applies only to forms, user displays, and form and user display groups.

Before you begin

Procedure

1. In the navigator, click **Global Displays**.
2. In the **Folders** list, select the resource type you want to work with.



Tip:

As more lists appear, select the appropriate resource until you find the one you want to work with.

3. Select the resource you want to duplicate, and then click **Duplicate**.
4. In the **Name** and **Description** fields, enter a name and description for the resource.
5. If applicable, click **Set**.

The Universal Browser appears where you can select the folder you want the duplicated display or group to be located.



Note:

When you duplicate a group, all of the displays that are part of that group are also duplicated.

6. Click **OK**.

Differences Between Duplicate Displays

The idea of duplicating is to make an exact copy of the original; however, in some cases there are some differences between duplicated forms and user displays.

The following scenarios describe the differences that occur with duplicated displays that were created in the Designer:

Scenario A

When you copy a display, the project name differs from the original because these names must be unique. The project name of the duplicated display is based on the display name specified with a number

appended to it, if necessary, to make it unique. The project name appears in the Project panel in the Designer.

Scenario B

When you duplicate a display, the .NET assembly that is generated displays a different name than the original. The assembly name matches the new project name.

The following scenarios describe the differences that occur with duplicated displays regardless of where they were created:

Scenario C

If a user display is duplicated to a location under the Forms folder, the base class is changed from `UserDisplayBase` to `FormBase` so that interaction with a workflow works as expected for a form.

Scenario D

If a form is duplicated to a location under the Displays folder, the base class is changed from `FormBase` to `UserDisplayBase`. If the original form has parameters or form events configured, you can still duplicate the form; however, the parameters and events are removed because user displays do not support them.

Exception to the Duplication Rule

After a display is duplicated, the original and duplicated item become independent of one another, where modifications to one do not affect the other. However, there is one exception to this rule.

If you create a form that points to an external .NET assembly file (created in another tool, such as Visual Studio), and then duplicate that form, both forms contain a reference to that assembly file. If you then upload a new version of that assembly file in the original form, then when the duplicate form is opened, the new version of the assembly file is also loaded for it.

However, this behavior is not specific to duplication. If you create two independent forms that use the same .NET assembly, and a new version of the assembly is uploaded to one of the forms, both forms will use the new version. Only the most recent version of the .NET assemblies are saved.

File Duplication in Forms and Displays

When forms or displays that were created in the Forms Designer are duplicated, the files in the project folder (in Windows Explorer) are also duplicated.

The duplicated files include the standard project files, as well as any additional files that may have been created if the form or display was edited in Visual Studio. Examples of files that will be duplicated are .sln, .suo, and files created by third party tools.

If you do not want to keep these files with the duplicated form/display, manually delete them from disk. The next time the project is saved, the files will no longer be saved with the duplicate. The .sln file will still point to the original project name, and therefore should be deleted from disk. These files will be recreated the next time the .csproj or .vbproj file is opened in Visual Studio.

Related information

[Form Development in Visual Studio \(on page 1224\)](#)

[Forms Development in Silverlight \(on page 1241\)](#)

Form Templates

Form Templates

Workflow provides samples of basic, predefined forms that are commonly used. You can use these forms in their basic format, or you can modify them to meet your needs.



Tip:

The forms in the Forms Library are available by adding the Forms Library .NET assembly (Proficy.Platform.FormsLibrary.dll) into the form and then selecting the class for the specific form. For more information, see *Basic Forms and Displays*.

Each form in the library has standard features. You can configure the parameters, as well as create expressions or binds.



Note:

The input parameters for the form are configured when the form is selected for a Form activity.

The following table describes the common, optional input parameters of the forms in the Forms Library:

Parameters	Data Type	Description
Comment Prompt	String	Prompt text displays above the comment text box.

Parameters	Data Type	Description
Comment Required	Boolean	If set to True, the Comment Prompt text box displays. If set to False, the text box is hidden and no comment can be entered.
ESignature Required	Boolean	If set to True, the User ID and Password text boxes display. If set to False, these text boxes are hidden.
Form Header	String	Enter a title for the form to display at the top of the form.
Form Header Visible	Boolean	If set to True, the header text displays. If set to False, the header text is hidden.

The following table describes the common output parameters of the forms in the Forms Library:



Note:

If the fields below are required (that is, set to True in the Inputs sections) and not filled in, an error message appears in the Feedback Zone.

Parameters	Data Type	Description
Comment	String	Displays the text entered in the Comment Prompt text box.
Password	String	Displays the text entered in the Password text box.
UserID	String	Displays the text entered in the User ID text box.

Related reference

[Form Design Environment \(on page 700\)](#)

[Basic Forms and Displays \(on page 1177\)](#)

Related information

[Form Development in Visual Studio \(on page 1224\)](#)

[Forms and User Displays Authoring Guide \(on page 699\)](#)

[Add the Work Request form to a workflow \(on page 1162\)](#)

Add the Work Request form to a workflow

Procedure

1. In the navigator, click **Global Displays**.



Note:

All areas that are compatible with the current resource appear. Green icons indicate a compatible view.

2. In the displays panel, select an editor.
The editor appears in the Workspace, displaying the resource information.
3. On the **Overview** tab, select the **Use existing** option.
4. Click **Set** next to the **.NET Assembly** field.
The **Open** dialog box appears.
5. Locate and select the .NET assembly file you want to use, and then click **Open**.
The .NET assembly file name appears and the **Class Name** drop-down list is populated with the classes specific to the selected .NET assembly.
6. From the **Class Name** drop-down list, select a class.
7. If applicable, in the **Additional Files** area, click **Add** to add any files that are required for the selected .NET assembly to work correctly.



Tip:

Click **Remove** to remove the selected file(s).

8. Click **Save**.
9. In the navigator, click **Workflow > Workflows**.
10. In the resource type list, select a resource.
11. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
12. If applicable, add a local subprocess to the workflow.
13. Insert and configure a **ReadSegmentRequirements** call method activity:

- a. Insert a call method activity.
- b. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
- c. Double-click **IProductionRuntime** to display the list of available call methods.
- d. Click **ReadSegmentRequirements**, and then click **OK**.
- e. On the activity configuration panel, in the Inputs section, click the drop-down arrow, and then select the segment details you want displayed in the work request form when it runs.
- f. Click the **Work Request** ellipsis button to open the **Universal Browser**, and then select the work request you want to associate with the call method.

- Or -

Bind the parameter to a call method return value or an input parameter; for example, from an event.

14. Insert and configure a form activity.

- a. Insert an activity.
- b. On the activity configuration panel, in the **General Settings** section, click the **Form** ellipsis button, and then select the form you want to use.
The **Inputs** section appears.
- c. If applicable, select the **Allow user to cancel form** check box.
- d. In the **Inputs** section, click the second drop-down arrow, and then select **Bind**.
The **Select Variable** dialog box appears.
- e. Expand the **ReadSegmentRequirement** activity, and then select the return value.
- f. Click **OK**.

15. Click **Save**.

When the form runs in the workflow, the information for the selected work request appears in the form.

Work Request Form

The Work Request form allows a user to view and configure work request data in a form in the Task List.

The Work Request form is part of the `Proficy.Platform.Forms.S95Forms.dll` .NET assembly and must use the **Proficy.Platform.Forms.S95Forms.SegmentRequirementsForm** class name.



Note:

The input parameters for the form are configured when the form is selected for a Form activity.

The Work Request form is used with the `ReadSegmentRequirements` call method. You must add the call method to the subprocess before you configure the form's input parameter.

General Settings

The following table describes parameters for this form:

Parameter	Data Type	Description
Form	String	Click the ellipsis button to open the Universal Browser , and then select the work request form you want to use.

Inputs

The following table describes parameters for this form:

Parameter	Data Type	Description
Segment Requirements	String	Click the drop-down arrow to open the Select Variable dialog box, and then select the return value of the associated ReadSegmentRequirements call method.

Check Boxes Form

The check boxes form allows you to create a form where multiple options can be selected by a user.

Inputs

The following table describes parameters for this form:

Parameters	Data Type	Description
Choices	String[]	Click the ellipsis button to open the String Collection Editor dialog box. Enter label names for each check box displayed in the form. <div data-bbox="1024 619 1424 888" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: The length of the array determines the number of check boxes displayed on the form. </div>
Prompt	String	Enter a heading to be displayed above the check box list.

Outputs

The following table describes parameters for this form:

Parameters	Data Type	Description
Return Choices	String[]	Displays the label names of the checked boxes.

Single Input Form

The single input form allows an operator to enter a single string/numeric value.

When a user enters a value, it is checked against the **Format Mask** input parameter and/or the High Limit and Low Limit input parameters. If the values match, then the form is ready to be submitted.

Inputs

The following table describes parameters for this form:

**Important:**

A value must be entered in the **Prompt** field.

Parameters	Data Type	Description
Default Value	String	<p>Specifies the initial value of the text field.</p> <p>You can also update this value by writing to it within a Form activity.</p> <div data-bbox="1024 659 1419 926" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note: Any new value must meet parameter requirements, as defined in this table.</p> </div>
Format Mask	String	<p>Specifies the mask for the text entered by the user, which is initially set to the default value.</p> <div data-bbox="1024 1106 1419 1373" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Tip: For more information on applying masks, see the Mask Properties in the MSDN Library.</p> </div>
High Limit	Double	<p>Enter the exclusive upper limit for a numeric value. Use this field if the data being retrieved is numerical and must fall within a range to advance the form.</p> <div data-bbox="1024 1644 1419 1890" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note: The value entered in the form must be up to the high limit value, not including the high limit val-</p> </div>

Parameters	Data Type	Description
Limits Visible	Boolean	<p> ue. For example, if the high limit is 100, the value cannot be higher than 99.</p> <p>If True, the high and low limits display in the form when it is running.</p>
Low Limit	Double	<p>Enter the exclusive lower limit for a numeric value. Use this field if the data being retrieved is numerical and must fall within a range to advance the form.</p> <p> Note: The value entered in the form must be above the low limit value, not including the low limit value. For example, if the low limit is 1, the value must be higher than 1.</p>
Prompt	String	Enter text to inform an operator to perform an action.

Outputs

The following table describes parameters for this form:

Parameters	Data Type	Description
Value as double	Double	Returns the value as type double.
Value as int	Int32	Displays the value text.
Value as String	String	Displays the text for the value when the form is submitted.

Multiple Input Form

The multiple input form allows you to create a data entry or data display form. Next to each label is a text box where the data can be entered.

Inputs

The following table describes parameters for this form:

Parameters	Data Type	Description
How Many Fields	Int32	Enter the number of fields to be displayed. <div data-bbox="1024 709 1421 1249" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: You can enter a maximum number of 10 fields into a form, starting from Item 0 and ending with Item 9. This value represents the total number of fields displayed in the final form layout regardless of the number of labels entered. </div>
Item 0 (through 9) Label	String	Enter the label name for the first form field.
Item 0 (through 9) Value	Int32	The default value in the text box is zero. If the form is not read-only, the user can change this value. If the form is read-only, the value entered by the form author displays.
Read Only	Boolean	If True, then all item values in the form are data display only. If False, then all item values are data entry items.

Outputs

The following table describes parameters for this form:

Parameters	Data Type	Description
Item 0 (through 9) Value Out	Int32	The value displayed or entered in the form.

Radio Buttons Form

The radio buttons form allows you to create a form with multiple options for a user to choose from.

Inputs

The following table describes parameters for this form:

Parameters	Data Type	Description
Default Index	Int	<p>Specifies the index of the initially selected radio button. If this number is less than zero, then the first is used. If this number is greater than the number of buttons, the last number is used.</p> <p>You can also update this value by writing to it within a Form activity.</p> <div data-bbox="1024 1314 1419 1535" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: Any new value must coincide with the available indexes.</p> </div>
Labels	String[]	<p>Click the ellipsis button to open the String Collection Editor dialog box. Enter label names for each radio button displayed in the form.</p>

Parameters	Data Type	Description
Prompt	String	Enter a heading to display above the radio button list.

 **Note:**
 The length of the array determines the number of buttons displayed on the form.

Outputs

The following table describes parameters for this form:

Parameters	Data Type	Description
Chosen Index	Int32	Displays the index of the selected radio buttons.
Chosen Label	String	Displays the label names of the selected radio buttons.

Subscription Data Form

The subscription data form allows you to retrieve a directory resource, when defined conditions are met, to advance a workflow task.

When you bind to a data item, an operator can perform an action based on the data item that was retrieved. The data item value is tied to mask and limits values. When the data item input parameter changes, the value is checked against the **Format Mask** input parameter and/or the High Limit and Low Limit input parameters. If the value(s) match(es), then the form is ready to be submitted.



Note:

These parameters are configured when the form is selected for a Form activity.

Inputs

The following table describes parameters for this form:

**Important:**

A value must be entered in the Data Item and Prompt fields.

Parameters	Data Type	Description
Auto Submit	Boolean	If True, there is no Submit button on the form when it is running. Therefore, when the input values match, the form submits automatically. If False, a Submit button appears on the form; it must be clicked to submit the form.
Data Item	N/A	Select a resource to use as a value to check against other input parameters.
Default Value	String	Initial value of the text field. This field can be used to introduce or clarify mask and limit information.
Format Mask	String	Enter a mask for the subscription data value.
High Limit	Double	If applicable, enter the exclusive upper limit for a numeric value. Use this field if the data being retrieved is numerical and must fall within a range to advance the form.

**Tip:**

For more information on applying masks, see the Mask Properties in the MSDN Library.

Parameters	Data Type	Description
		<div data-bbox="1024 260 1419 709" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;">  Note: The value entered in the form must be up to the high limit value, not including the high limit value. For example, if the high limit is 100, the value cannot be higher than 99. </div>
Limits Visible	Boolean	If True, the high and low limits display in the form when it is running.
Low Limit	Double	If applicable, enter the exclusive lower limit for a numeric value. Use this field if the data being retrieved is numerical and must fall within a range to advance the form.
Prompt	String	<div data-bbox="1024 1178 1419 1577" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;">  Note: The value entered in the form must be above the low limit value, not including the low limit value. For example, if the low limit is 1, the value must be higher than 1. </div> <p data-bbox="1019 1598 1414 1724">Enter text to provide information to an operator to perform an action.</p>

Outputs

The following table displays parameters for this form:

Parameters	Data Type	Description
Value as double	Double	Displays the value text.
Value as int	Int32	Displays the value text.
Value as String	String	Displays the text for the value when the form is submitted.

Text Array Forms

The text array form allows you to create a table with two columns; one column has labels for each row, and the other column allows for data entry.

When a user enters a value, it is checked against the **Format Mask** input parameter and/or the High Limit and Low Limit input parameters. If the value(s) match(es), then the form is ready to be submitted.

Inputs

The following table describes parameters for this form:

Parameters	Data Type	Description
Format Masks	String[]	Click the ellipsis button to open the String Collection Editor dialog box. Enter a mask for each value.
High Limits	Double[]	Click the ellipsis button to open the Double Collection Editor dialog box. If applicable, create the exclusive upper limit for a numeric value. Use this field if the data being retrieved is numerical and must fall within a range to advance the form.

 **Tip:**
For more information on applying masks, see the Mask Properties in the MSDN Library.

Parameters	Data Type	Description
Input Table Labels	String[]	<div data-bbox="1024 262 1425 709" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-bottom: 10px;">  Note: The value entered in the form must be up to the high limit value, not including the high limit value. For example, if the high limit is 100, the value cannot be higher than 99. </div> <p>Click the ellipsis button to open the String Collection Editor dialog box. Enter label names for each row in the form.</p>
Input Table Values	String[]	<p>Click the ellipsis button to open the String Collection Editor dialog box. Enter an initial value for each row in the form.</p>
Label Header	String	<p>Enter a heading for the left hand label column.</p>
Low Limits	Double[]	<p>If applicable, enter the exclusive lower limit for a numeric value. Use this field if the data being retrieved is numerical and must fall within a range to advance the form.</p> <div data-bbox="1024 1522 1425 1806" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: The value entered in the form must be above the low limit value, not including the low limit value. For example, if the </div>

Parameters	Data Type	Description
Value Header	String	<div data-bbox="1023 262 1421 367" style="border: 1px solid #0070C0; border-radius: 5px; padding: 5px;">  low limit is 1, the value must be higher than 1. </div> <p>Enter a heading for the data entry column (left hand).</p>

Outputs

The following table describes parameters for this form:

Parameters	Data Type	Description
Output Table Values	String[]	Displays the array of values entered.

Text Grid Forms

The text grid form allows you to create a table with a grid pattern. A user can enter or display a two dimensional table of values.

When a user enters a value, it is checked against the **Format Mask** input parameter. If the values match, then the form is ready to be submitted. Inputs

The following table describes parameters for this form:

Parameters	Data Type	Description
Format Masks	String[]	<p>Click the ellipsis button to open the String Collection Editor dialog box. Enter a mask for each value.</p> <div data-bbox="1023 1512 1421 1774" style="border: 1px solid #0070C0; border-radius: 5px; padding: 5px;">  Tip: For more information on applying masks, see the Mask Properties in the MSDN Library. </div>
Input Table Headers	String[]	Click the ellipsis button to open the String Collection Editor dia-

Parameters	Data Type	Description
		log box. Enter heading names for each grid column in the form.
		 Note: The length of the array determines the number of columns.
Input Table Values	String[]	Click the ellipsis button to open the String Collection Editor dialog box. Enter an initial value for each grid cell in the form.
Justifications	String[]	The format justifications of each column; that is, left, right, or center.

Outputs

The following table describes parameters for this form:

Parameters	Data Type	Description
Output Table Values	String[]	Displays the array of values entered.

Yes/No Form

The yes or no form allows a user to select either yes or no in a form.

Inputs

The following table describes parameters for this form:

Parameters	Data Type	Description
Default Choice	Boolean	If set to True, the default choice in the form is Yes. If set to False, the default choice in the form is No.

Parameters	Data Type	Description
No	String	You can also update this value by writing to it within a Form activity. Enter a label name for this radio button displayed in the form.
Prompt	String	Enter a heading to display above the radio button list.
Yes	String	Enter a label name for this radio button displayed in the form.

Outputs

The following table describes parameters for this form:

Parameters	Data Type	Description
Choice	Boolean	Displays the user's selected choice between yes or no.

Basic Forms and Displays

Basic Forms and Displays

You can design a display a form backed by Microsoft's .NET technology or a web form backed by HTML using a graphical interface.

The **Overview** tab in the **Display Editor** is your starting point for creating forms or user displays. You can select an existing assembly or web page for a display or form.

Attribute	Value	Description
Name	Required; user-defined up to 50 characters	Specifies a display name for a resource.
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

Graphical Designer Options

Attribute	Value	Description
Use the Designer tab to create new client content	User-defined	Select this option to create a form or user display using the Designer .

Create a form or user display with a design editor

You can create a form or display with a built-in graphical design editor in Workflow.

Procedure

1. In the navigator, click **Global Displays**.



Note:

All areas that are compatible with the current resource appear. Green icons indicate a compatible view.

2. In the **Displays** panel, click **Display Editor**.
3. On the **Overview** tab, select Use the Designer tab to create new rich client content.
4. Click the **Designer** tab.
5. If applicable, in the dialog box that appears, in the **Project Name** field, enter a project name, or accept the displayed project name.



Note:

The project name is the same as the resource name. Changing the project name does not affect the resource name.

6. From the **Language** list, select the programming language you want to use for code behind, and then click **OK**.
7. Design and create your form or user display using the **Designer**.
8. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms
title bar	to the server for WPF forms

Related information

[Use a form in a workflow \(on page 1181\)](#)

Modify a form or user display

Before you begin

This procedure applies to all forms and user displays.

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. In the **Forms** list, select the resource you want to work with.



Note:

All areas that are compatible with the current resource appear. Green icons indicate a compatible view.

3. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
4. If applicable, on the **Overview** tab, click **Edit**, and then change the name and description of the form.
5. Make the required changes to your form or user display.
6. Click **Save**.

Preview a form or user display in the Global Viewer

You can preview forms and user displays that were developed in the **Designer**, as well as web pages and web content.

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. In the **Forms** list, select the resource you want to work with.



Note:

All areas that are compatible with the current resource appear. Green icons indicate a compatible view.

3. To view your form or display, in the **Displays** panel, click **Global Viewer**.

Configure forms to view during run time after they have been completed or acquired

Depending on your access permissions and client configuration, you can select either My Task List or Task Management view.

About this task

You can view completed or acquired forms in the Task List if they are configured to support this.

Procedure

1. Use the following table to perform tasks.

To view...	Go to step...
task history of a completed form	2
recently completed tasks during runtime	8

2. In the navigator, click **Workflow**.
3. In the **Folders** list, add a workflow or select an existing workflow. For more information, see **Add a resource to the workflow model**.
4. Click the **Tracking Configuration** tab.
5. Select **Enable Tracking**.
6. Select **Track at the Process and Subprocess Level**.
7. Click **Save**.
8. In the **Models** list, click **Global Displays**.
9. In the **Folders** list, add a form or select an existing form. For more information, see **Add a Global Displays resource**.
10. In the displays panel, click **Display EditorDesignerDocument TreeProperties**.
11. Select the **ReloadableForm** check box.
12. Optionally, to display read-only data in a completed or acquired (reloaded) state, use the *IsReloaded* variable in your form. This Boolean value indicates whether a form is being viewed for the first time (False), or if it has been reloaded (True).



Note:

You cannot submit a form that is loaded in this state.

The following is an example of a common purpose for using *IsReloaded*:

- a. From the Document Tree FormBase, select a user interface element that you want to view in the Task List.
 - b. From the Properties tab, click the **IsReadOnly** check box, and then from the drop-down arrow, select **Expression**. The Expression dialog box appears.
 - c. Click the **Expression Variables** tab.
 - d. Expand Controls, expand FormBase, and then double-click **IsReloaded** to add it to the expression field.
13. Save the form or user display.

Click Save on the...	To save...
toolbar	locally for WPF forms; to the server for Silverlight forms.
title bar	to the server for WPF forms.

14. You can view forms in several ways. From the Task List of a running workflow:
- Click the **Task History** tab, and then click the **Input** button. Recently completed task information is shown. To view previously run forms, from the Form drop-down menu, select a form.
 - Click the **Tasks** tab, and then from the **Task Steps** list, select a previously completed step of a running task.
 - You can view a currently running form that has been acquired by someone else.

**Note:**

A completed task cannot be re-executed; therefore, the **Submit** and **Cancel** buttons are not available.

Add a Global Displays resource

Procedure

1. In the navigator, click **Global Displays**.
2. In the **Folders** list, click **Displays** to add a display, or **Forms** to add a form.
3. Click **Add Display** or **Add Form**, depending on the resource you are adding.
4. In the **Name** and **Description** fields, enter a name and description for the resource.
5. Click **OK**.

Use a form in a workflow

Use this procedure to add a form to a workflow.

Before you begin



Note:

You must have already added a pre-existing form or created a custom form before you can add it to a workflow.

About this task

You can add a form that was created in Visual Studio or a custom form that was developed using the Forms Designer. If the form has input and output parameters, it will typically have a preceding activity to give it input, and a subsequent activity to receive its output. These other activities may themselves be forms, which can be useful for testing.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the resource type list, select a resource.
4. Click **Add**.
5. In the **Name** and **Description** fields, enter a name and description for the resource.
6. Click **OK**.
7. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
8. Click the **Workflow Editor** tab.
9. Drag and drop a **Local Subprocess** into the workflow and double-click it, or double-click an existing subprocess.
10. Insert a Form activity.
11. To activate the form at run time, click the **Overview** tab, and then click **Enable**.



Note:

Only activated subprocesses are available for workflows.

12. Click **Save**.
13. To test the viability of this workflow, see *Start and end a test debugging session*.

Set the equipment context in a form

You can dynamically update the equipment context by using a client method on a WPF form.

Prerequisites

[Add a client method to a form or display \(on page 1129\)](#)

About this task**Note:**

You cannot set a WPF form's equipment context for use in the Web Task List.

Procedure

1. From the navigator, click **Global Displays > Forms**.
2. Add a WPF form, or from the **Forms** list, select an existing WPF form.
3. Click **Display Editor > Designer**.
4. Click the **MainForm.xaml** tab.
5. Click the **Toolbox** tab.
6. Select the control that you want to add an equipment context to, and then drag and drop it into the **Designer**.
7. To configure the control's appearance, click the **Properties** tab.
8. Click the **Client Methods** tab.
9. Click **Add**.
A client method appears in the list.
10. Select the client method, and then click the **Properties** tab to configure the method.
11. In the **Source** field, click the ellipsis button.
The **Configure Method** dialog box appears.
12. Configure the properties, as required.

**Note:**

To view the full list of properties, select the **Advanced** check box.

13. To set the equipment context in a workflow Task List, click **SetEquipmentContext**, and then click **Finish**.

**Note:**

All manual settings are overridden if a form in the Task List uses the **Set Equipment Context** client method to assign equipment.

14. In **Properties** panel, in the **Equipment** field:
 - Click the ellipsis button to link to a piece of equipment, or
 - Click the drop-down arrow and bind to an input parameter.
15. **Optional:** In the **EventTriggers** field, click **Add**.
The **Event Triggers** dialog box appears.

16. **Optional:** From the **Controls** drop-down list, expand the button that you want to add an event trigger to, select **Click**, and then click **OK**.
17. Click **Save**.
18. To use the functionality of this form, add it to a workflow.
 - a. Insert and configure a Form activity.
 - b. In the **Form** field, click the ellipsis button to add the form that contains the equipment context button.
 - c. Click **Save**.
19. **Optional:** To prevent changing context information during run time, you can hide or disable the buttons on the Task List.
 - a. Go to My Task List Editor.
 - b. In the **Visible** column:
 - i. To hide the **Context** button, and all information related to computer, user, and equipment context, clear the **Task List Context** check box.
 - ii. To allow users read-only access to context information, clear the **Equipment Context Editing** check box.
 - iii. Click **Save**.

Web-Based Forms and Displays

Web-based Forms and Displays

You can use the **Designer** to create web-based forms.

The following table describes the common, optional input parameters of the web-based forms and displays:

Attribute	Value	Description
Name	Required; user-defined up to 50 characters	Specifies a display name for a resource.
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

Graphical Designer Options

The **Overview** tab in the **Display Editor** is your starting point for creating forms or user displays. You can design a display or form backed by Microsoft's .NET technology or a web form backed by HTML using a graphical interface. Alternatively, you can select an existing assembly or web page for a display or form.



Note:

You can import and export custom HTML forms and simple web forms.

Attribute	Value	Description
Use the Designer tab to create web content	User-defined	Select this option to create a form or user display using Silverlight for Silverlight forms.
Use the Designer tab to create a simple web form	User-defined	Select this option to create an HTML form or user display. <div data-bbox="1024 932 1419 1402" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note: These forms can be used during run time in the following locations only:</p> <ul style="list-style-type: none"> • Workflow Task List (regular) • Mobile Task List • Vision Task List </div>
Use a web page	User-defined	Select this option to specify a URL for a user display or form that is based on a web page.

Developing a web form for a workflow

This tutorial shows how you can design, develop, and use a web-based form, then add it to a simple workflow for use by an operator working in the Task List.

Before you begin

The following procedures indicate the basic steps involved in creating a simple workflow with a web form.

- Log in and launch Workflow client
- Create and configure a web form with appropriate labels and operator controls



Tip:

Text fields can be used to display read-only information, such as headings or titles on your form. Text boxes can be used for processing operator input.

- Create and configure a workflow
- Add a form to a workflow
- Add activities to a workflow to demonstrate binding user input to a form
- Add instructions for operators
- Create a schedule to automatically start a workflow
- Test the workflow in the Task List

About this task



Note:

For a video tutorial of this topic, use a browser to open <http://www.youtube.com>, and then search for `GE Proficiency Workflow - Developing a Web Form for a Workflow`.

Procedure

1. Log in to Workflow.
 - a. Launch Workflow client using either the **Start** menu or desktop icon.
 - b. On the main Workflow screen, click  .
The **Workflow Login** dialog box appears.
 - c. **Optional:** In the **Authentication Type** section, select the type appropriate to your user profile.
 - d. In the **User Name** field, enter your user name.
 - e. In the **Password** field, enter your password.
 - f. Click **OK**.
2. Create a web form resource.
 - a. From the navigator, click **Global Displays > Forms**.
 - b. Click **Add Form**.
 - c. In the **Name** field, enter a name, such as `My First Form`.

- d. In the **Description** field, enter clarifying information about the web form resource, such as
`This is my first web form.`
 - e. Click **OK**, or press `ENTER`.
 - f. In the **Displays** panel, click **Display Editor**.
 - g. On the **Overview** tab, select **Use the Designer tab to create a simple web form**.
 - h. Click **Save**.
3. Design a form by adding a read-only text field to the web form.
 - a. Click the **Designer** tab.
 - b. In the **Form Fields** section, click the **Text** form control to add the field to your form.

**Tip:**

You can add multiple text fields, as well as other form controls. Each text control is labelled and numbered in the **Name** field in sequence; for example, *Text1*, *Text2*, and so on.

To delete a field, select it, and then click **Delete**.

The **Layout** area displays the basic configuration for your form.

- c. In the **Properties** section, you can configure the form fields.
 - i. In the **Name** field, enter a unique identifier by modifying the existing text. For example, enter `Text Control Name`. For example, enter `Text Control Name`.
 - ii. In the **Text** field, enter content that is displayed to the operator and provides more information about the form. For example, enter `Form Title`.
 - iii. In the **Size** field, from the drop-down list, select a text size.
 - iv. In the **Alignment** field, from the drop-down list, select an alignment value.
4. Design a form by creating a text box to the web form.
 - a. In the **Form Fields** section, click the **Text Box** form control to add the field to your form.
The **Layout** area displays the basic configuration for your form.
 - b. In the **Properties** section, you can configure the form fields.
 - i. In the **Name** field, enter a unique identifier by modifying the existing text. For example, enter `Text Box Control Name`.
 - ii. In the **Label** field, enter descriptive text on the form control that is displayed to the operator. For example, enter `Text Box with Status Message`.
 - iii. From the **Type** drop-down list, ensure **Text** is selected.

No further configuration is required.

- c. Click **Save**.
5. To view your form or display, in the **Displays** panel, click **Global Viewer**.
 6. Create a workflow resource.
 - a. In the navigator, click **Workflow > Workflows**.
 - b. Click **Add Workflow**.
 - c. In the **Name** field, enter a name, such as `My First Workflow`.
 - d. In the **Description** field, enter clarifying information about the workflow resource, such as `This is my first workflow`.
 - e. Click **OK**, or press `ENTER`.
 7. Configure the workflow resource.
 - a. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
 - b. Click the **Workflow Editor** tab.
 - c. To assign a task in the workflow, on the **General** tab, in the **Personnel** field, click the ellipsis button.
 - d. In the **Universal Browser**, click **Personnel > People**, and then select your user name.
 - e. Click **OK**, or press `ENTER`.
 - f. Drag and drop a **Local Subprocess** into the workflow and double-click it, or double-click an existing subprocess.
 - g. From the subprocess configuration panel, click **Edit**.
 - h. In the **Name** field, enter a name, such as `WebFormStep`, and then click **OK**.
 - i. Click **Save**.
 - j. Click the **Overview** tab.
 - k. In the **Status** section, click **Enable**.
 8. Add a form to the workflow.
 - a. Click the **Workflow Editor** tab.
 - b. From the workspace, double-click the subprocess, `WebFormStep`.
 - c. From the **Standard Activities** panel, drag and drop a **Form** activity into the subprocess.
 - d. From the activity configuration panel, click **Edit**.
 - e. In the **Name** field, enter a name, such as `WebFormActivity`, and then click **OK**.
 - f. From the activity configuration panel, in **General Settings**, click the ellipsis button next to the **Forms** field.

The **Universal Browser** appears.

- g. In the **Universal Browser**, click **Global Displays > Forms**, and then select the form that you created, `My First Form`.
 - h. Click **OK**, or press `ENTER`.
 - i. Click **Save**.
9. Add activities to the subprocess to display form information to and request input from an operator. You can bind user input from the form's text box control (step 4) to workflow variables. For example, you can add a **Status Message** activity to your subprocess in order to deliver a message to your operator regarding a particular task, and request input.
- a. From the **Standard Activities** panel, drag and drop a **Status Message** activity into the subprocess after the **Form** activity.
 - b. From the activity configuration panel, in **General Settings**, click the drop-down arrow next to the **Message** field, and then select **Expression** to bind the message in your form.
 - c. In the **Enter Expression** field, enter the following expression: `"User entered:"+`
 - d. Click the **Workflow Variables** tab.
 - e. Expand the **Form** activity, `WebFormActivity`, that you added to the subprocess.
 - f. To complete the expression, double-click the text box value, `Text Box Control Name.Value`, that you created in step 4.
You have created the following expression: `"User entered:"+{WebFormActivity}.{Text Box Control Name.Value}`
 - g. Click **OK**.
 - h. From the **Standard Activities** panel, drag and drop a **Delay** activity into the subprocess after the **Status Message** activity.
 - i. In the **Timeout** field, select the **seconds** section, and then use the arrows to set a 5-second delay.
 - j. Click **Save**.
10. Add work instructions to the workflow.
- a. Click the **Work Instructions** tab.
 - b. From the **Task/Task Steps** list, select the task step, `WebFormStep`.

- c. In the **Work Instructions** area, enter instructions or insert an image.
 - d. Click **Save**.
11. Create a workflow schedule to start the workflow automatically, and then activate it so that it is available in production.
 - a. In the navigator, click **Workflow > Schedules**.
 - b. Click **Add Schedule**.
 - c. In the **Name** field, enter a name, such as `My Schedule`.
 - d. In the **Description** field, enter clarifying information about the resource, such as `This is my first schedule`.
 - e. Click **OK**, or press `ENTER`.
 - f. In the **Displays** panel, click **Workflow Management > Workflow Editor > Overview**.
 - g. In the **Select Workflow** section, click **Browse**.
The **Universal Browser** appears.
 - h. Select the workflow that you created in step 6, `My First Workflow`
 - i. In the **Start Task Filter** section, select the **Is schedule visible in Start Task window** check box.
 - j. Click **Save**.
 - k. In the **Status** section, click **Enable**.
12. Run and test the workflow.

- a. From the title bar, click the Task List icon,  .
- b. Click **Start Task**.
- c. Select the schedule that you created in step 11, `My Schedule`.
- d. Click **Start**.
The task that you created is now visible in the Task List. You can enter data into the form, view work instructions, and submit changes.

What to do next**Tip:**

For Vision users, click the **TASKS** tab to see pending workflows.

Create a simple web form with images

You can use the **Designer** to create basic HTML forms with simple images, such as a logo.

About this task**Note:**

You can create HTML forms for use only in the Workflow client Task List and in the Mobile Task List.

Procedure

1. In the navigator, click **Global Displays**.
2. From the **Forms** folder, add a new form, or select a form from the **Forms** list.
3. In the **Displays** panel, click **Display Editor**.
4. On the **Overview** tab, select **Use the Designer tab to create a simple web form**.
5. Click **Save**.
6. Click the **Designer** tab.
7. In the **Form Fields** section, click the **Image** form control to add the field to your form.

You can add multiple image fields, as well as other form controls. Each image is labelled and numbered in the **Name** field in sequence; for example, *Image1*, *Image2*, and so on.

The **Layout** area displays the basic configuration for your form.
8. **Optional:** To move fields around the **Layout** area, select the field to drag and drop it to a different location within the panel.
9. **Optional:** To delete a field, select it, and then click **Delete**.
10. In the **Properties** section, you can configure the form fields.
 - a. In the **Name** field, enter a unique identifier by modifying the existing text.

You can view the name of the field property when you add a Form activity to a workflow in the **Workflow Editor**. You can then bind a value to this property.
 - b. In the **Label** field, enter descriptive text on the form control that is displayed to the operator.
 - c. In the **URL** field, enter the absolute path of the image's location.
 - d. In the **Alignment** field, from the drop-down list, select an alignment value.

- Left
- Center
- Right

e. In the **Height** field, enter a whole number to represent the number of pixels for the height of the image.



Note:

If no value is specified, the default dimensions are used.

f. In the **Width** field, enter a whole number to represent the number of pixels for the width of the image.



Note:

If no value is specified, the default dimensions are used.

11. Click **Save**.

Create a user display or form from a web page

You can use the **Designer** to create web-based forms.

Procedure

1. In the navigator, click **Global Displays**.



Note:

All areas that are compatible with the current resource appear. Green icons indicate a compatible view.

2. Select one of the following options.

To create a...	Do this...
Form	Add a new form, or select a form from the Forms list.
Display	Add a new display, or select a user display from the Displays list.

3. In the **Displays** panel, click **Display Editor**.

4. On the **Overview** tab, select **Use a web page**, and then enter a valid URL.
5. To view your form or display, in the **Displays** panel, click **Global Viewer**.
6. Click **Save**.

Create a simple web form with generic text fields

You can use the **Designer** to create basic HTML forms with generic, read-only text fields that you can use to add headings or titles to you forms.

About this task



Note:

You can create HTML forms for use only in the Workflow client Task List and in the Mobile Task List.

Procedure

1. In the navigator, click **Global Displays**.
2. From the **Forms** folder, add a new form, or select a form from the **Forms** list.
3. In the **Displays** panel, click **Display Editor**.
4. On the **Overview** tab, select **Use the Designer tab to create a simple web form**.
5. Click **Save**.
6. Click the **Designer** tab.
7. In the **Form Fields** section, click the **Text** form control to add the field to your form.

You can add multiple text fields, as well as other form controls. Each text control is labelled and numbered in the **Name** field in sequence; for example, *Text1*, *Text2*, and so on.

The **Layout** area displays the basic configuration for your form.
8. **Optional:** To move fields around the **Layout** area, select the field to drag and drop it to a different location within the panel.
9. **Optional:** To delete a field, select it, and then click **Delete**.
10. In the **Properties** section, you can configure the form fields.
 - a. In the **Name** field, enter a unique identifier by modifying the existing text.



Note:

You can view the name of the field property when you add a Form activity to a workflow in the **Workflow Editor**. You can then bind a value to this property.

- b. In the **Text** field, enter content that is displayed to the operator and provides more information about the form.

- c. In the **Size** field, from the drop-down list, select a text size.
 - Normal
 - Small
 - Medium
 - Large
- d. In the **Alignment** field, from the drop-down list, select an alignment value.
 - Left
 - Center
 - Right

11. Click **Save**.

Create a simple web form with text boxes

You can use the **Designer** to create basic HTML forms with text boxes.

About this task



Note:

You can create HTML forms for use only in the Workflow client Task List and in the Mobile Task List.

Procedure

1. In the navigator, click **Global Displays**.
2. From the **Forms** folder, add a new form, or select a form from the **Forms** list.
3. In the **Displays** panel, click **Display Editor**.
4. On the **Overview** tab, select **Use the Designer tab to create a simple web form**.
5. Click **Save**.
6. Click the **Designer** tab.
7. In the **Form Fields** section, click the **Text Box** form control to add the field to your form.

You can add multiple text fields, as well as other form controls. Each text control is labelled and numbered in the **Name** field in sequence; for example, *TextBox1*, *TextBox2*, and so on.

The **Layout** area displays the basic configuration for your form.
8. **Optional:** To move fields around the **Layout** area, select the field to drag and drop it to a different location within the panel.
9. **Optional:** To delete a field, select it, and then click **Delete**.
10. In the **Properties** section, you can configure the form fields.

- a. In the **Name** field, enter a unique identifier by modifying the existing text.



Note:

You can view the name of the field property when you add a Form activity to a workflow in the **Workflow Editor**. You can then bind a value to this property.

- b. In the **Label** field, enter descriptive text on the form control that is displayed to the operator.
- c. In the **Type** drop-down list, select one of the following options.
- Text, and then go to step 11.
 - Number, and then go to step 12.
 - Password, and then go to step 13.
 - Hidden, and then go to step 14.
11. To allow an operator to enter text only into the text box, configure the related properties.
- a. From the **Required** drop-down list, select one of the following options:
- True, to ensure the operator enters a value in the form. If no value is entered, the form cannot be submitted.
 - False, to give the operator the option to enter a value in the form. If no value is entered, there is no effect on form submission.
- b. **Optional:** In the **Value** field, enter predefined text in the field for the operator to view.
- c. **Optional:** In the **Maximum Characters** field, enter a whole number to indicate the number of character space available to operators during run time in the related field.
12. To allow an operator to enter numbers only into the text box, configure the related properties.
- a. From the **Required** drop-down list, select one of the following options:
- True, to ensure the operator enters a value in the form. If no value is entered, the form cannot be submitted.
 - False, to give the operator the option to enter a value in the form. If no value is entered, there is no effect on form submission.
- a. **Optional:** In the **Minimum Value** field, set the fewest number of characters required for the operator to enter into the text box; and/or
- b. **Optional:** In the **Maximum Value** field, set the greatest number of characters required for the operator to enter into the text box.
13. To ensure an operator enters a password into the text box, configure the related properties.

- a. From the **Required** drop-down list, select one of the following options:
 - True, to ensure the operator enters a value in the form. If no value is entered, the form cannot be submitted.
 - False, to give the operator the option to enter a value in the form. If no value is entered, there is no effect on form submission.
 - b. **Optional:** In the **Maximum Characters** field, enter a whole number to indicate the number of character space available to operators during run time in the related field.
14. To hide the text box from the operator, configure the related property.
- a. **Optional:** In the **Value** field, enter predefined text in the field for the operator to view.
15. Click **Save**.

Create a simple web form with a text area

You can use the **Designer** to create basic HTML forms with an area for operators to add text.

About this task

**Note:**

You can create HTML forms for use only in the Workflow client Task List and in the Mobile Task List.

Procedure

1. In the navigator, click **Global Displays**.
2. From the **Forms** folder, add a new form, or select a form from the **Forms** list.
3. In the **Displays** panel, click **Display Editor**.
4. On the **Overview** tab, select **Use the Designer tab to create a simple web form**.
5. Click **Save**.
6. Click the **Designer** tab.
7. In the **Form Fields** section, click the **Text Area** form control to add the field to your form.

You can add multiple text fields, as well as other form controls. Each text control is labelled and numbered in the **Name** field in sequence; for example, *TextArea1*, *TextArea2*, and so on.

The **Layout** area displays the basic configuration for your form.
8. **Optional:** To move fields around the **Layout** area, select the field to drag and drop it to a different location within the panel.
9. **Optional:** To delete a field, select it, and then click **Delete**.
10. In the **Properties** section, you can configure the form fields.

- a. In the **Name** field, enter a unique identifier by modifying the existing text.

**Note:**

You can view the name of the field property when you add a Form activity to a workflow in the **Workflow Editor**. You can then bind a value to this property.

- b. In the **Label** field, enter descriptive text on the form control that is displayed to the operator.
- c. From the **Required** drop-down list, select one of the following options:
 - True, to ensure the operator enters a value in the form. If no value is entered, the form cannot be submitted.
 - False, to give the operator the option to enter a value in the form. If no value is entered, there is no effect on form submission.
- d. **Optional:** In the **Value** field, enter predefined text in the field for the operator to view.
- e. **Optional:** In the **Number of Lines** field, enter a whole number to indicate the number of available lines available to the operator in the text area during run time.
- f. **Optional:** In the **Maximum Characters** field, enter a whole number to indicate the number of character space available to operators during run time in the related field.

11. Click **Save**.

Create a simple web form with check boxes

You can use the **Designer** to create basic HTML forms with check boxes.

About this task

**Note:**

You can create HTML forms for use only in the Workflow client Task List and in the Mobile Task List.

Procedure

1. In the navigator, click **Global Displays**.
2. From the **Forms** folder, add a new form, or select a form from the **Forms** list.
3. In the **Displays** panel, click **Display Editor**.
4. On the **Overview** tab, select **Use the Designer tab to create a simple web form**.
5. Click **Save**.

6. Click the **Designer** tab.
7. In the **Form Fields** section, click the **Check Box** form control to add the field to your form.
You can add multiple text fields, as well as other form controls. Each text control is labelled and numbered in the **Name** field in sequence; for example, *CheckBox1*, *CheckBox2*, and so on.
The **Layout** area displays the basic configuration for your form.
8. **Optional:** To move fields around the **Layout** area, select the field to drag and drop it to a different location within the panel.
9. **Optional:** To delete a field, select it, and then click **Delete**.
10. In the **Properties** section, you can configure the form fields.
 - a. In the **Name** field, enter a unique identifier by modifying the existing text.

**Note:**

You can view the name of the field property when you add a Form activity to a workflow in the **Workflow Editor**. You can then bind a value to this property.

- b. In the **Label** field, enter descriptive text on the form control that is displayed to the operator.
 - c. From the **Required** drop-down list, select one of the following options:
 - True, to ensure the operator enters a value in the form. If no value is entered, the form cannot be submitted.
 - False, to give the operator the option to enter a value in the form. If no value is entered, there is no effect on form submission.
 - d. From the **Auto Check** drop-down list, select one of the following options:
 - True, to display a check box to the operator that is already marked.
 - False, to display a check box to the operator that is clear.
11. Click **Save**.

Create a simple web form with radio buttons

You can use the **Designer** to create basic HTML forms with radio buttons.

About this task

**Note:**

You can create HTML forms for use only in the Workflow client Task List and in the Mobile Task List.

Procedure

1. In the navigator, click **Global Displays**.
2. From the **Forms** folder, add a new form, or select a form from the **Forms** list.
3. In the **Displays** panel, click **Display Editor**.
4. On the **Overview** tab, select **Use the Designer tab to create a simple web form**.
5. Click **Save**.
6. Click the **Designer** tab.
7. In the **Form Fields** section, click the **Radio Buttons** form control to add the field to your form.
You can add multiple text fields, as well as other form controls. Each text control is labelled and numbered in the **Name** field in sequence; for example, *RadioButtons1*, *RadioButtons2*, and so on.
The **Layout** area displays the basic configuration for your form.
8. **Optional:** To move fields around the **Layout** area, select the field to drag and drop it to a different location within the panel.
9. **Optional:** To delete a field, select it, and then click **Delete**.
10. In the **Properties** section, you can configure the form fields.

- a. In the **Name** field, enter a unique identifier by modifying the existing text.



Note:

You can view the name of the field property when you add a Form activity to a workflow in the **Workflow Editor**. You can then bind a value to this property.

- b. In the **Label** field, enter descriptive text on the form control that is displayed to the operator.
- c. From the **Required** drop-down list, select one of the following options:
 - True, to ensure the operator enters a value in the form. If no value is entered, the form cannot be submitted.
 - False, to give the operator the option to enter a value in the form. If no value is entered, there is no effect on form submission.
- d. **Required:** In the **Options** field, enter one or more values that is viewable to an operator during run time as a drop-down list.
- e. Click **Save** to populate the **Default Option** drop-down list.
- f. **Optional:** From the **Default Option** drop-down list, select an entry from your **Options** list to indicate which item you want the operator to interact with first.

The default option is **(none)**. If you do not change this value, there is no option selected in the run-time form.

11. Click **Save**.

Create a simple web form with combo boxes

You can use the **Designer** to create basic HTML forms with combo boxes.

About this task

**Note:**

You can create HTML forms for use only in the Workflow client Task List and in the Mobile Task List.

Procedure

1. In the navigator, click **Global Displays**.
2. From the **Forms** folder, add a new form, or select a form from the **Forms** list.
3. In the **Displays** panel, click **Display Editor**.
4. On the **Overview** tab, select **Use the Designer tab to create a simple web form**.
5. Click **Save**.
6. Click the **Designer** tab.
7. In the **Form Fields** section, click the **Combo Box** form control to add the field to your form.
You can add multiple text fields, as well as other form controls. Each text control is labelled and numbered in the **Name** field in sequence; for example, *ComboBox1*, *ComboBox2*, and so on.
The **Layout** area displays the basic configuration for your form.
8. **Optional:** To move fields around the **Layout** area, select the field to drag and drop it to a different location within the panel.
9. **Optional:** To delete a field, select it, and then click **Delete**.
10. In the **Properties** section, you can configure the form fields.
 - a. In the **Name** field, enter a unique identifier by modifying the existing text.

**Note:**

You can view the name of the field property when you add a Form activity to a workflow in the **Workflow Editor**. You can then bind a value to this property.

- b. In the **Label** field, enter descriptive text on the form control that is displayed to the operator.

- c. **Required:** In the **Options** field, enter one or more values that is viewable to an operator during run time as a drop-down list.
- d. Click **Save** to populate the **Default Option** drop-down list.
- e. **Optional:** From the **Default Option** drop-down list, select an entry from your **Options** list to indicate which item you want the operator to interact with first.

The default option is (**none**). If you do not change this value, there is no option selected in the run-time form.

11. Click **Save**.

Create a simple web form with a list box

You can use the **Designer** to create basic HTML forms with a list box that allows an operator to select multiple entries at the same time.

About this task



Note:

You can create HTML forms for use only in the Workflow client Task List and in the Mobile Task List.

Procedure

1. In the navigator, click **Global Displays**.
2. From the **Forms** folder, add a new form, or select a form from the **Forms** list.
3. In the **Displays** panel, click **Display Editor**.
4. On the **Overview** tab, select **Use the Designer tab to create a simple web form**.
5. Click **Save**.
6. Click the **Designer** tab.
7. In the **Form Fields** section, click the **List Box** form control to add the field to your form.
You can add multiple text fields, as well as other form controls. Each text control is labelled and numbered in the **Name** field in sequence; for example, *ListBox1*, *ListBox2*, and so on.
The **Layout** area displays the basic configuration for your form.
8. **Optional:** To move fields around the **Layout** area, select the field to drag and drop it to a different location within the panel.
9. **Optional:** To delete a field, select it, and then click **Delete**.
10. In the **Properties** section, you can configure the form fields.

- a. In the **Name** field, enter a unique identifier by modifying the existing text.



Note:

You can view the name of the field property when you add a Form activity to a workflow in the **Workflow Editor**. You can then bind a value to this property.

- b. In the **Label** field, enter descriptive text on the form control that is displayed to the operator.
- c. From the **Required** drop-down list, select one of the following options:
 - True, to ensure the operator enters a value in the form. If no value is entered, the form cannot be submitted.
 - False, to give the operator the option to enter a value in the form. If no value is entered, there is no effect on form submission.
- d. **Required:** In the **Options** field, enter one or more values that is viewable to an operator during run time as a drop-down list.
- e. Click **Save** to populate the **Default Option** drop-down list.
- f. **Optional:** From the **Default Option** drop-down list, select an entry from your **Options** list to indicate which item you want the operator to interact with first.

The default option is **(none)**. If you do not change this value, there is no option selected in the run-time form.

11. Click **Save**.

Save the HTML code for customization

You can save the HTML code from a simple web form into a separate file, and then customize it to meet your specific requirements.

About this task

You must save your form in order to save the form code.

Procedure

1. In the navigator, click **Global Displays**.
2. From the **Forms** folder, add a new form, or select a form from the **Forms** list.
3. In the **Displays** panel, click **Display Editor**.
4. Click the **Designer** tab.
5. Click **Save As**.

The **Save Simple Web Form** dialog box appears.

- Find the location to where you want to save your file, and then in the **File name** field, accept the default name, or provide a new name for your file.



Note:

The file name defaults to the name of your form, with .html extension.

- Click **Save**.

Text Fields in Web Forms

You can use the **Designer** to create basic HTML forms with generic, read-only text fields that you can use to add headings or titles to your forms.

Properties

Attribute	Value	Description
Name	User-defined alphanumeric string (A-Z, a-z, 0-9)	Specifies a unique identifier that is displayed in a workflow's Form activity. You can then bind properties associated with the selected form field.
Text	User-defined alphanumeric string (A-Z, a-z, 0-9)	Displays content that provides information to the operator about the form. For example, you can use this property to add a title to your form.
Size	User-defined	Specifies the size of the text in the form, such as normal, small, medium, or large.
Alignment	User-defined	Specifies the location of the text in the form, such as left, right, or center.

Images in Web Forms

You can use the **Designer** to create basic HTML forms with simple images, such as a logo.

Properties

Attribute	Value	Description
Name	User-defined alphanumeric string (A-Z, a-z, 0-9)	Specifies a unique identifier that is displayed in a workflow's Form activity. You can then bind properties associated with the selected form field.
Label	User-defined alphanumeric string (A-Z, a-z, 0-9)	Specifies descriptive text on the form control that is displayed to the operator.
URL	User-defined alphanumeric string (A-Z, a-z, 0-9)	Specifies the web address where the image is located. <div data-bbox="1024 852 1419 1121" style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: This path must be accessible by the machine that the operator is working on during run time. </div>
Alignment	User-defined	Specifies the location of the text in the form, such as left, right, or center.
Height	User-defined numeric value	Specifies the height in pixels.
Width	User-defined numeric value	Specifies the width in pixels.

Text Boxes in Web Forms

You can use the **Designer** to create basic HTML forms with text boxes.

Standard Text Box Properties

You can create a standard text box.

Attribute	Value	Description
Name	User-defined alphanumeric string (A-Z, a-z, 0-9)	Specifies a unique identifier that is displayed in a workflow's Form

Attribute	Value	Description
		activity. You can then bind properties associated with the selected form field.
Label	User-defined alphanumeric string (A-Z, a-z, 0-9)	Specifies descriptive text on the form control that is displayed to the operator.
Actual Tool Type: Text	User-defined	Specifies the format of the text box that appears in the form.
Required	Boolean	<p>Select one of the following options:</p> <ul style="list-style-type: none"> • True, to ensure the operator enters a value in the form. If no value is entered, the form cannot be submitted. • False, to give the operator the option to enter a value in the form. If no value is entered, there is no effect on form submission.
Value	User-defined	Optional; Specifies a predefined text value inside the form field that is visible to operators during run time.
Maximum Characters	User-defined numeric value	Optional; Specifies the number of character spaces available in the form field.

Number Text Box Properties

You can create text boxes that contain numeric values.

Attribute	Value	Description
Name	User-defined alphanumeric string (A-Z, a-z, 0-9)	Specifies a unique identifier that is displayed in a workflow's Form activity. You can then bind properties associated with the selected form field.
Label	User-defined alphanumeric string (A-Z, a-z, 0-9)	Specifies descriptive text on the form control that is displayed to the operator.
Type: Number	User-defined	Specifies the format of the text box that appears in the form.
Required	Boolean	<p>Select one of the following options:</p> <ul style="list-style-type: none"> • True, to ensure the operator enters a value in the form. If no value is entered, the form cannot be submitted. • False, to give the operator the option to enter a value in the form. If no value is entered, there is no effect on form submission.
Value	User-defined alphanumeric string (A-Z, a-z, 0-9)	Optional; Specifies the number of character spaces available in the form field.
Minimum Value	User-defined numeric value	Optional; Specifies the fewest number of characters allowed in the text box. You can choose to set this value on its own, with the Maximum Value field, or not at all.
Maximum Value	User-defined numeric value	Optional; Specifies the greatest number of characters allowed in

Attribute	Value	Description
		the text box. You can choose to set this value on its own, with the Minimum Value field, or not at all.

Password Text Box Properties

You can create text boxes that are used to allow operators to input passwords in forms.

Attribute	Value	Description
Name	User-defined alphanumeric string (A-Z, a-z, 0-9)	Specifies a unique identifier that is displayed in a workflow's Form activity. You can then bind properties associated with the selected form field.
Label	User-defined alphanumeric string (A-Z, a-z, 0-9)	Specifies descriptive text on the form control that is displayed to the operator.
Type: Password	User-defined	Specifies the format of the text box that appears in the form.
Required	Boolean	Select one of the following options: <ul style="list-style-type: none"> • True, to ensure the operator enters a value in the form. If no value is entered, the form cannot be submitted. • False, to give the operator the option to enter a value in the form. If no value is entered, there is no effect on form submission.
Maximum Characters	User-defined numeric value	Optional; Specifies the number of character spaces available in the form field.

Hidden Text Box Properties

You can create text boxes that are used for a form's system functions and are not visible to operators.

Attribute	Value	Description
Name	User-defined alphanumeric string (A-Z, a-z, 0-9)	Specifies a unique identifier that is displayed in a workflow's Form activity. You can then bind properties associated with the selected form field.
Label	User-defined alphanumeric string (A-Z, a-z, 0-9)	Specifies descriptive text on the form control that is displayed to the operator.
Type: Hidden	User-defined	Specifies the format of the text box that appears in the form.
Value	Required; User-defined up to 50 characters	Optional; Specifies a predefined text value inside the form field that is visible to operators during run time.

Text Areas in Web Forms

You can use the **Designer** to create basic HTML forms with an area for operators to add text.

Properties

Attribute	Value	Description
Name	User-defined alphanumeric string (A-Z, a-z, 0-9)	Specifies a unique identifier that is displayed in a workflow's Form activity. You can then bind properties associated with the selected form field.
Label	User-defined alphanumeric string (A-Z, a-z, 0-9)	Specifies descriptive text on the form control that is displayed to the operator.

Attribute	Value	Description
Required	User-defined Boolean	Select one of the following options: <ul style="list-style-type: none"> • True, to ensure the operator enters a value in the form. If no value is entered, the form cannot be submitted. • False, to give the operator the option to enter a value in the form. If no value is entered, there is no effect on form submission.
Value	User-defined alphanumeric string (A-Z, a-z, 0-9)	Optional; Specifies a predefined text value inside the form field that is visible to operators during run time.
Number of Lines	User-defined numeric value	Specifies the quantity of lines available in the text area.
Maximum Characters	User-defined numeric value	Optional; Specifies the number of character spaces available in the form field.

Check Boxes in Web Forms

You can use the **Designer** to create basic HTML forms with check boxes.

Properties

Attribute	Value	Description
Name	User-defined alphanumeric string (A-Z, a-z, 0-9)	Specifies a unique identifier that is displayed in a workflow's Form activity. You can then bind properties associated with the selected form field.

Attribute	Value	Description
Label	User-defined alphanumeric string (A-Z, a-z, 0-9)	Specifies descriptive text on the form control that is displayed to the operator.
Required	User-defined; Boolean	<p>Select one of the following options:</p> <ul style="list-style-type: none"> • True, to ensure the operator enters a value in the form. If no value is entered, the form cannot be submitted. • False, to give the operator the option to enter a value in the form. If no value is entered, there is no effect on form submission.
Auto Check	User-defined; Boolean	<p>Select one of the following options:</p> <ul style="list-style-type: none"> • True, to display a check box to the operator that is already marked with a check mark. • False, to display a check box to the operator that is clear. The operator is responsible for selecting the box in the form at run time.

Radio Buttons in Web Forms

You can use the **Designer** to create basic HTML forms with radio buttons.

Properties

Attribute	Value	Description
Name	User-defined alphanumeric string (A-Z, a-z, 0-9)	Specifies a unique identifier that is displayed in a workflow's Form activity. You can then bind properties associated with the selected form field.
Label	User-defined alphanumeric string (A-Z, a-z, 0-9)	Specifies descriptive text on the form control that is displayed to the operator.
Required	User-defined; Boolean	<p>Select one of the following options:</p> <ul style="list-style-type: none"> • True, to ensure the operator enters a value in the form. If no value is entered, the form cannot be submitted. • False, to give the operator the option to enter a value in the form. If no value is entered, there is no effect on form submission.
Options	User-defined; String	Required; Specifies the values for each form field line that are viewable to an operator during runtime as a drop-down list.
Default Option	User-defined; Integer	<p>Optional; From the list of options for the control, select the one that you want the operator to interact with first.</p> <p>The default option is (none). If you do not change this value,</p>

Attribute	Value	Description
		there is no option selected in the run-time form.

Combo Boxes in Web Forms

You can use the **Designer** to create basic HTML forms with combo boxes.

Properties

Attribute	Value	Description
Name	User-defined alphanumeric string (A-Z, a-z, 0-9)	Specifies a unique identifier that is displayed in a workflow's Form activity. You can then bind properties associated with the selected form field.
Label	User-defined alphanumeric string (A-Z, a-z, 0-9)	Specifies descriptive text on the form control that is displayed to the operator.
Options	User-defined; String	Required; Specifies the values for each form field line that are viewable to an operator during run time as a drop-down list.
Default Option	User-defined; String	Optional; From the list of options for the control, select the one that you want the operator to interact with first.

List Boxes in Web Forms

You can use the **Designer** to create basic HTML forms with a list box that allows an operator to select multiple entries at the same time.

Properties

Attribute	Value	Description
Name	User-defined alphanumeric string (A-Z, a-z, 0-9)	Specifies a unique identifier that is displayed in a workflow's Form activity. You can then bind properties associated with the selected form field.
Label	User-defined alphanumeric string (A-Z, a-z, 0-9)	Specifies descriptive text on the form control that is displayed to the operator.
Required	User-defined; Boolean	<p>Select one of the following options:</p> <ul style="list-style-type: none"> • True, to ensure the operator enters a value in the form. If no value is entered, the form cannot be submitted. • False, to give the operator the option to enter a value in the form. If no value is entered, there is no effect on form submission.
Options	User-defined; String	Required; Specifies the values for each form field line that are viewable to an operator during runtime as a drop-down list.
Default Option	N/A	<p>Optional; From the list of options for the control, select the one that you want the operator to interact with first.</p> <p>The default option is (none). If you do not change this value,</p>

Attribute	Value	Description
		there is no option selected in the run-time form.

Saving HTML Code

You can save the HTML code from a simple web form into a separate file, and then customize it to meet your specific requirements.

Property	Value	Description
Save As	N/A	Click this button to open the Save Simple Web Form dialog box, and then save the opened form's HTML code into an HTML file.

Installing and Viewing Sample Custom Web Forms

A sample custom web form is provided when you install Workflow 2.2 SP1 and later. The sample form contains examples of some form controls, including data grids, that are supported. It features an example of how to use a DataTable table with hard-coded values to populate a data grid in the form.

Installation Location

The sample form files are available in the following directory:

```
<32-bit installdir>\Program\Applications\Workflow\Samples\Sample Custom Web Form.
```

This sample contains:

- The HTML for a web form that makes use of a *jQuery* data grid.
- The *jQuery* libraries needed to use the data tables in the form.
- A modified `ActivityExtensions.dita` file that makes the JSON (JavaScript Object Notation) libraries available to a workflow.
- A Workflow Project that contains:
 - A workflow that demonstrates the conversion between a JSON string and a DataTable using the Newtonsoft.JSON library, as well as creating and using the data table in a Code activity. The workflow uses the JSON string to pass data for a grid into the form.
 - A schedule to start the sample workflow.
 - A form object that references the sample form.

Viewing the Sample Form in the Task List

1. The contents of `Form Web Page`, including the subfolders, must be hosted on a web server.
2. The file, `tasklistform.min.js` (in `<32-bit installdir>\Program\Applications\Workflow\SDK`), must also be hosted.
3. Update `SampleCustomForm.html` to use the correct paths for script and style imports. Use fully qualified server names (for more information, see *Requirements for Creating Custom Web Forms*).
4. Update the following file, which can be found in the installation directory appropriate to your system architecture (32- or 64-bit), `<installdir>\Program\Data\Projects\SOAPProject\AppServers\Workflow\Workflow\ActivityExtensions.dita`, to include the `NewtonSoft.JSON` library (see sample in `<installdir>\Program\Applications\Workflow\Samples\ActivityExtensions`).
5. Restart your Workflow application server.
6. Deploy the `Sample Custom Form.zip` project (in `<installdir>\Proficy\Proficy\Workflow\Program\Applications\Workflow\Samples\Sample Custom Web Form\Workflow Project`).
7. Update the form `Sample Custom Form (HTML)` to use the correct URL for your web server (for example, `https://<MyServer>:8443/SampleForm/SampleCustomForm.html`).
8. Enable the workflow and schedule named `Sample - Show Custom Form`.
9. Start the `Sample - Show Custom Form` schedule and view the task in the Task List.

Data Grids in the Sample Custom Form

Data grids have been styled in the sample custom form using the *jQuery* user interface. For additional themes, go to the following web site: <http://jqueryui.com/>. These themes can be substituted by hosting them and updating the link in the sample form to point to the appropriate CSS.

**Note:**

When creating other custom forms, you can use whichever style you prefer.

Related reference

[Requirements for Creating Custom Web Forms \(on page 1215\)](#)

Requirements for Creating Custom Web Forms

You can create custom web forms, and then bind, build an expression, or set a literal value against the input and output parameters in a workflow.

Form Library

All custom web forms must include the form library, which is contained in `tasklistform.min.js` via a script tag (for example, `<script src="https://<Insert Host Name Here>/tasklistform.min.js"></script>`). This file is located in `<32-bit installdir>\Program\Applications\Workflow\SDK`. You can also find a file that provides information about calling the required functions that allow forms to work in the Task List.

Custom Form Template

The file, `FormTemplate.html`, provides information about calling the required functions that allow forms to work in the Task List. This file is located in `<32-bit installdir>\Program\Applications\Workflow\SDK`.

Communicating with the Task List

In order for your form to send and receive data from the Task List, your form must contain the appropriate calls to the form library. The form must call:

- **formLoaded()** when it is ready to receive input data; for example, in the *onloaded* event of your body element
- **submitForm()** to perform a submission back to workflow; for example, in the *onsubmit* handler of your form element
- **cancelForm()** when it wishes to signal a cancellation of the form; for example, in the *onclick* handler of a cancel button)

For examples of how these calls are performed, see the included sample form.

Using Custom Web Forms in the Workflow Task List

In order for custom HTML forms to function properly in the regular Task List, you must ensure the following line is added at the start of the **<head>** section of your form.

```
<meta http-equiv="X-UA-Compatible" content="IE=Edge,chrome=1">
```

Hosting Sample Files

You must update the linked CSS files and script elements in `Sample Custom Form.html` to point to the location where the files are hosted. To do so, replace **<Insert Host Name Here>** in the following code with the full host name.

```

<link rel="stylesheet" type="text/css" media="screen" href="https://<Insert Host Name
Here>/SampleForm/css/jquery-ui.css"/>

<link rel="stylesheet" type="text/css" media="screen" href="https://<Insert Host Name
Here>/SampleForm/css/ui.jqgrid.css"/>

<script src="https://<Insert Host Name Here>/SampleForm/js/jquery-1.11.0.min.js"></script>
<script src="https://<Insert Host Name Here>/SampleForm/js/i18n/grid.locale-en.js"></script>
<script src="https://<Insert Host Name Here>/SampleForm/js/jquery.jqGrid.src.js"></script>
<script src="https://<Insert Host Name Here>/SampleForm/js/jquery-ui.js"></script>
<script src="https://<Insert Host Name Here>/SampleForm/js/tasklistform.min.js"></script>

```

Accessing Custom Web Forms

Customers who create custom web forms must also provide secure web hosting in order for these forms to be accessible by Workflow. As a result, you must use an HTTPS address. When you configure a custom web form, use only this HTTPS address in the URL field.



Note:

The HTTPS configuration for custom HTML forms is also applicable to Workflow task client.

To use a custom HTML form with the Web HMI Task List, Web HMI must be configured to allow connections to the custom form host.

The custom form will fail to load if the reverse proxy is not configured. Use the proxy whitelist to accomplish the configuration.

1. Open the C:\Program Files\Proficy\ProficyWebServer\ReverseProxy\serverConfig.json file on the Web HMI Server.
2. Edit the whitelist element to include the custom form host. e.g., "whitelist" : ["https://myformhost:port"],
3. Restart the "GE Proxy" windows service.

Form Events in Web-based Forms

For more information on using form events, see *Event Handlers for Web-based Forms*.

Requirements for Binding Input and Output Parameters

- There must be exactly one <form> element in the custom HTML.
- All elements must have a unique **Name** attribute.

**Note:**

Exception for radio buttons: All radio buttons in a radio group have the same name.

- Labels: The **For** attribute must be set to the **ID** of the corresponding element.

**Note:**

Exception for radio buttons: Labels for radio groups have the **Name** property set to the same **Name** of each radio button. A label element must be present to ensure that the radio button options can be modified from a running workflow.

- Authentication: For custom web forms to work properly, there must be no authentication on the server hosting these forms.
- Modifying forms: If you make changes to the custom form, you must save it in order for the changes to work.
- To tell the host which methods to call when passing in the initial workflow inputs and when collecting the form outputs to return, the form must call **addInputParameterFunction** and **addOutputParameterFunction**. This call is handled automatically for most basic HTML elements, such as <p>, <input>, <textarea>, <select>, and .
- Inputs:
 - Add an input parameter function with the string name **<elementName>.InitialValue** for each control input that you want to handle.
 - In the sample form, the *jqGrid* populates using a JSON representation of a table:

```
if (addInputParameterFunction) {
    addInputParameterFunction('jqGrid.InitialValue', populateForm);
}
```

- Outputs:
 - Add an output parameter function with the name **<elementName>.Value** for each control output that you want to provide. Returns a string.
 - In the sample form, a JSON representation of the table is returned:

```
if (addOutputParameterFunction) {
    addOutputParameterFunction(collectFormOutput);
}
```

Supported Binding Elements

Control	Tag	Element
Text	<p>	<ul style="list-style-type: none"> • value
Image		<ul style="list-style-type: none"> • URL
Input	<input>	<ul style="list-style-type: none"> • label • value
Selection	<select>	<ul style="list-style-type: none"> • label • options • selected index • selected value
Radio Buttons	<input>	<ul style="list-style-type: none"> • label • options • selected index • selected value
List Box	<select multiple>	<ul style="list-style-type: none"> • all selection options, and • multi-select outputs
Others	N/A	<ul style="list-style-type: none"> • value (provided as string) <div data-bbox="1105 1150 1419 1822" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note: The custom form must provide a mechanism for applying the provided values to the control and returning a string as output. For more information, see Requirements for Binding Input and Output Parameters above.</p> </div>

Hosting Custom Web Forms

You can use custom forms that are hosted on a separate domain.

HTTP Headers

To load custom web forms, you must:

1. Create a secure web host.
2. Add the following HTTP header to each host response: `Access-Control-Allow-Origin`.
3. Configure the HTTP header value for each host response to include the domain information where you will be hosting the custom forms.
4. Add a second HTTP header to each host response: `Access-Control-Allow-Headers` with the value `authorization,content-type,pragma,cache-control`.

Using IIS to Host Custom Web Forms



Note:

This section assumes that you have a knowledge of web administration, and that you have already set up a web application and created a location containing a custom web form.

1. Start the Internet Information Services (IIS) Manager application.
2. Select the folder that contains the form.
3. From the **IIS** section in the central panel, double-click **HTTP Response Headers**.
4. From the **Actions** panel, click **Add**.
5. In the **Name** field, enter **Access-Control-Allow-Origin**.
6. In the **Value** field, enter the host where the Task List will be hosted. For example, for the Vision web server, enter `https://mywebserver`.
7. Click **OK**.
8. To add another header, from the **Actions** panel, click **Add**.
9. In the **Name** field, enter **Access-Control-Allow-Headers**.
10. In the **Value** field, enter `authorization,content-type,pragma,cache-control`.
11. Click **OK**.

Existing Assembly Options

Existing Assembly Options

The **Overview** tab in the **Display Editor** is your starting point for creating forms or user displays. You can design a display or form backed by Microsoft's .NET technology or a web form backed by HTML using a graphical interface. Alternatively, you can select an existing assembly or web page for a display or form.

Attribute	Value	Description
Name	Required; user-defined up to 50 characters	Specifies a display name for a resource
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.

Graphical Designer Options

Attribute	Value	Description
Use an existing assembly	User-defined	Select this option to specify an existing .NET assembly and class name.

Pre-existing File Options

The following table describes the required fields to create a form or user display from an existing assembly file.

Attribute	Description
.NET Assembly	Click Set to locate and select the .NET assembly that contains the display you want to add. This field is available only when the Use an existing assembly check box is selected.
Class Name	Specifies the class name of the display contained by the selected .NET assembly. This field is

Attribute	Description	
		available only when the Use an existing assembly check box is selected.
Additional Files		<p>Specifies any additional files required for the selected .NET assembly to work correctly.</p> <div data-bbox="1024 575 1419 1871" style="border: 1px solid #ccc; border-radius: 10px; background-color: #fff9c4; padding: 10px;"> <p> Important:</p> <ul style="list-style-type: none"> • When you specify an additional file, a copy is made and saved to the database. If you save and navigate away from the current form or user display, the absolute path to the file location changes to the database location when you open the form or user display again. <p>If you make changes to the local version of the file (that is, the original version), including moving or deleting it, those changes are not applied to the database ver-</p> </div>

Attribute	Description	
		 <p>sion. In order for the changes to be applied, you must remove the database version of the file and then add it again.</p> <ul style="list-style-type: none"> • If you test a form in the Designer that contains additional files, an error message may appear. This error occurs only when you are working in the Designer. The error does not occur in the run time environment.
Add		Click to browse and select the files that are required for the selected .NET assembly to work correctly.
Remove		Click to remove the selected files.

Related reference

[Basic Forms and Displays \(on page 1177\)](#)

Add pre-existing forms and user displays to Workflow

This procedure is used for forms and user displays that were created outside of the Designer, in a tool such as Visual Studio.

Procedure

1. In the navigator, click **Global Displays**.



Note:

All areas that are compatible with the current resource appear. Green icons indicate a compatible view.

2. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
3. On the **Overview** tab, select **Use an existing assembly**.
4. Click **Set** next to the **.NET or Silverlight Assembly** field.
The **Open** dialog box appears.
5. Locate and select the .NET assembly file you want to use, and then click **Open**.
The .NET assembly file name appears and the **Class Name** drop-down list is populated with the classes specific to the selected .NET assembly.
6. From the **Class Name** drop-down list, select a class.
7. If applicable, in the **Additional Files** area, click **Add** to add any files that are required for the selected .NET assembly to work correctly.



Tip:

Click **Remove** to remove the selected file(s).

8. Click **Save**.

Form Development in Visual Studio

Form Development in Visual Studio

A form can be used as an activity in a workflow subprocess to provide a user interface (UI) that works with a workflow task step.



Note:

This section discusses the basics of building WPF-based forms with VB.Net with some attention given to WinForm-based forms.

Introduction

A form can display data or allow you to interact with a running process, or both. VB.Net projects with user controls are used to build the forms. A pre-built base form is provided to reduce the amount of coding required. The base form handles implementing the communication interfaces to and from a workflow.

References are used to add classes to the coding project. Overrides are used to override the inputs and outputs of the base form and to allow the passing of inputs to and from a workflow into your form code. A ValidateForm method allows code to be used to verify the data in the form and provide a message to the end user if the data is not valid.

These are the core aspects of building custom forms. This section also includes advanced topics for reading and writing data internally to and from the form and also to subscribe to events within the form.

Supported UI Technology Types

The following are the UI technology types supported in Workflow:

- Windows Forms (WinForms)
- Web
- Windows Presentation Foundation (WPF)

This section deals primarily with WPF-based forms that are developed in Visual Studio 2005 or 2008 with VB.Net. WinForm-based development is similar; however, there are several steps that differ. Additional notes are provided for those steps where WinForm-based development differs.

Prerequisites

To build forms, you need either:

- Visual Studio 2008 (for WPF forms or WinForms)
- Visual Studio 2005 (for WinForms only).

Features and Capabilities of Forms

- Inherit from an existing BaseForm to minimize coding
- Pass data into and out of the form
- Validation capability
- Read data internally from other available data service providers to the form, as well as from Workflow

- Write data from the form to other data service providers, as well as outputs to Workflow
- Forms can be set up to published events from other service providers, as well as Workflow events

The Interfaces

In order to communicate with the Workflow client in a standard way, a form should implement the `IForm` interface. You can implement the interfaces yourself, but the easiest way to achieve this is to extend a base class.

Available Forms

A base form is provided, which implements the communication interfaces. To use the base form, you must inherit it, and then override three methods (to input, output, and validate data). It is located in the Program folder of the installation directory, and is called `Proficy.Platform.Samples.Forms.BaseForm.dll`.

Sample forms are also included. You can copy one of the sample forms and modify it to create a new form. The samples are available for download.



Note:

If you are creating a form based on WinForms, then use the base form located in `Proficy.Platform.Samples.Forms.BaseFormVB.dll`.

Related reference

[Form Design Environment \(on page 700\)](#)

Related information

[Forms and User Displays Authoring Guide \(on page 699\)](#)

[Form Templates \(on page 1160\)](#)

[Forms Development in Silverlight \(on page 1241\)](#)

Extend the base form

Procedure

1. In Visual Studio, create a new VB project form using the WPF User Control Library or Class Library project templates.

**Note:**

If you are creating a form based on WinForms, use the "User Windows Forms Control Library" (which is called "Windows Control Library" in Visual Studio 2005) or "Class Library" project templates.

2. Delete the Class1.vb file from the project.
3. Add a new WPF user control to the project using the "User Control (WPF)" file template.

**Note:**

If you are creating a form based on WinForms, use the "User Control" file template.

4. In the References section of the project, browse to the Program folder of the Workflow installation directory and add references to the following:

- Proficy.Platform.Core.DisplayFramework.dll
- Proficy.Platform.Core.ProficySystem.dll
- Proficy.Platform.Samples.Forms.Baseform.dll

If you are using WPF, add the following .NET references if they are not already added to your project:

- PresentationCore
- PresentationFramework
- WindowsBase

If you are using WinForms, add the following .NET reference if it is not already added to your project:

- System.Windows.Forms

5. Add the following imports to include these classes in your project.

- System
- System.Collections.Generic
- System.Windows
- System.Windows.Controls
- Proficy.Platform.Samples.Forms.Base

These imports are required in order to resolve the types used by BaseForm.

**Note:**

If you are creating a form based on WinForms rather than WPF, you should import `System.Windows.Controls` instead of `System.Windows.Forms`. The other imports mentioned above remain the same.

6. Change the class to inherit from the `BaseForm`; that is, change the `.vb` file to contain the following

`Inherits` statement:

```
Public Class UserControl1
```

```
Inherits Proficiency.Platform.Samples.Forms.Base.BaseForm
```

You must also modify the `.xaml.vb` file to derive from that class, as well, by adding the namespace of the `BaseForm` and changing the beginning and ending tags to use it. When you are finished, the `.xaml` file should look as follows:

```
<base:BaseForm x:Class="UserControl1"
xmlns:base="clr-namespace:Proficiency.Platform.Samples.Forms.Base;assembly=Proficiency.Platform.Samples.Forms.BaseForm"
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" Width="300" Height="300">
<Grid>
</Grid>
</base:BaseForm>
```

**Note:**

If you are creating a form based on WinForms, then your form should inherit from `BaseFormVB.Proficiency.Platform.Samples.Forms.Base.VB.BaseFormVB`.

An error displays because the other partial class does not have the same `Inherits` line.

Press `Shift+Alt+PF10`, click the red underscore below `BaseFormVB`, and then select the *Change class "UserControl1" to inherit from "BaseFormVB.Proficiency.Platform.Samples.Forms.Base.VB.BaseFormVB"* link. The `Inherits` line is added to the partial class.

**Note:**

Do not give your class a namespace.

7. Switch to the Designer View and create the user interface elements for the form, such as text boxes, labels, and list boxes.

**Note:**

If you are creating a form based on WinForms, set the Autosize property on the user control to True. This allows the form to automatically size to the window pane size in the Task List.

8. Switch to the Code View and paste in Regions code for Private Variables, InputParameters, OutputParameters, and Overrides. There are three functions in the Overrides region:
 - TransferDataIn: This function gets called prior to displaying the form. It allows for the transfer of data from the input parameters into the display components.
 - ValidateForm: This function gets called to validate the form prior to attempting to return parameters. This function gets called before TransferDataOut.
 - TransferDataOut: This function gets called prior to returning the outputs from the form. It allows for the transfer of data from the display components and the setting the output fields.
- For more information, see *Overriding the Base Form Methods* and *Declaring Parameters*.
9. Build the project.
 10. When the project builds successfully into a DLL, add it to Workflow as a new form. See *Add pre-existing forms and user displays to Workflow*.

Related reference

[Overriding the Base Form Methods \(on page 1232\)](#)

[Declaring Parameters \(on page 1231\)](#)

Related information

[Add pre-existing forms and user displays to Workflow \(on page 1223\)](#)

Add pre-existing forms and user displays to Workflow

This procedure is used for forms and user displays that were created outside of the Designer, in a tool such as Visual Studio.

Procedure

1. In the navigator, click **Global Displays**.



Note:

All areas that are compatible with the current resource appear. Green icons indicate a compatible view.

2. In the **Displays** panel, select an editor.
The editor appears in the workspace, displaying the resource information.
3. On the **Overview** tab, select **Use an existing assembly**.
4. Click **Set** next to the **.NET or Silverlight Assembly** field.
The **Open** dialog box appears.
5. Locate and select the .NET assembly file you want to use, and then click **Open**.
The .NET assembly file name appears and the **Class Name** drop-down list is populated with the classes specific to the selected .NET assembly.
6. From the **Class Name** drop-down list, select a class.
7. If applicable, in the **Additional Files** area, click **Add** to add any files that are required for the selected .NET assembly to work correctly.



Tip:

Click **Remove** to remove the selected file(s).

8. Click **Save**.

Add custom submit, cancel, and enable to a form using the Designer

The form can be configured so that the Workflow Task List Submit and Cancel buttons are not visible when the form is displayed in the Task List. You can also ensure that a form is only ready for submission after the appropriate operator interaction by enabling/disabling the **Submit** button.

Procedure

1. From the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. In the **Displays** panel, click **Display Editor**.
4. In the **Document Tree** tab, select the root of the document tree.
5. In the root element's property panel, clear the **IsSubmitVisible** and **IsCancelVisible** check boxes to hide the Task List Submit and Cancel buttons.

6. **Optional:** To wait for operator interaction before enabling the **Submit** button, from the code behind, set the **IsSubmitEnabled** property to false to disable the **Submit** button.
7. To add code-designed forms to a workflow, see *Configure a Form activity*.

Enable a form Submit button after operator interaction

You can ensure that the form's **Submit** button becomes active after an operator has entered data into the form. Using the **IsSubmitEnable** property prevents an operator from submitting a form prematurely because it does not contain data, or because the data is not valid.

Procedure

1. In either the Designer or Visual Studio:
 - a. Open or create an existing form.
 - b. From `MainForm.xaml`, open the `MainPage.xaml.cs` file.
2. In the code behind, add the following custom code: `IsSubmitEnabled = false`
The **Submit** button will now become enabled after the operator performs an action.
3. To add code-designed forms to a workflow, see *Configure a Form activity*.

Declaring Parameters

A form has input data that it processes into output data, even though it may not have a user interface. This section discusses the input and output parameters, which are passed to and from the client using the `IForm` interface.

The base form implements this interface; therefore your variables need to be declared only with the specific attribute of `InputProperty` or `Outputproperty`, as follows:

```
<Global.Proficiency.Platform.Samples.Forms.Base.InputProperty("Name", "DisplayName ", "Description")> _
Protected _input As String = "Default Value"

<Global.Proficiency.Platform.Samples.Forms.Base.OutputProperty("Name", " DisplayName ", "Description")> _
Protected _output As String
```

The following is an example of input parameters:

```
#Region "InputParameters"

<Global.Proficiency.Platform.Samples.Forms.Base.InputProperty("QEventIn", "QEvent", "QEvent")> _
Protected QEventIn As String

<Global.Proficiency.Platform.Samples.Forms.Base.InputProperty("SelectLabelIn", "SelectLabel", "SelectLabel")> _
Protected SelectLabelIn As String

<Global.Proficiency.Platform.Samples.Forms.Base.InputProperty("tblEventsIn", "Table Events", "Table Events")> _
```

```

Protected tblEventsIn As DataTable
#End Region

Here is an example of Output Parameters:

#Region "OutputParameters"

    <Global.Proficy.Platform.Samples.Forms.Base.OutputProperty _
    ("SelectedRowsOut", "SelectedEvents", "Events selected using the form.")> _

    Protected SelectedEventsOut As DataTable

    <Global.Proficy.Platform.Samples.Forms.Base.OutputProperty _
    ("ActionOut", "Action", "Action")> _

    Protected ActionOut As String

#End Region

```

With the `InputProperty` attribute, the variable automatically receives its initial value from the client, if it is bound. With the `OutputProperty` attribute, the value of the variable is automatically passed to the Client for binding after the form has been submitted.

The arguments for constructing the attributes are Name, DisplayName, and Description. The Name is used internally (that is, it is never displayed) and it must be unique among both input and output properties. The DisplayName is the name that users see in the **Workflow Editor**, in the Input and Output pane, and must also be unique. The Description appears in the **Workflow Editor** under the DisplayName, for example, in tooltips.

Related information

[Form Development in Visual Studio \(on page 1224\)](#)

Overriding the Base Form Methods

Overrides are used to override the inputs and outputs of the base form and to allow you to pass inputs to and from a workflow into your form code.

The three base form methods to be overridden are, in order of execution:

TransferDataIn()

This is where you write the code that sets up the user interface according to the input parameters, or any other initialization. This method is called after the input parameters have received their value, but before the form is displayed.

ValidateForm(ByRef errorMessage As String) As Boolean

This is where you write the code that validates the form after the Submit button is clicked. If there is an error, `errorMessage` is set to display in a dialog box, and `False` is returned; otherwise `True` is returned. This method is called after the Submit button is clicked.

TransferDataOut()

This is where you write the code that sets the values of the output parameters, according to the current state of the user interface. This method is called only after `ValidateForm` is returned as `True`.

The following sample form shows a list of radio buttons and an optional comment box:

```

Import System

Imports System.Collections.Generic

Imports System.Windows

Imports System.Windows.Controls

Imports Proficy.Platform.Samples.Forms.Base

Public Class ChoicesVB

    Inherits Proficy.Platform.Samples.Forms.Base.BaseForm

    #Region "Private Variables"

        Private _buttons As List(Of RadioButton)

    #End Region

    #Region "Overrides"

        ' This function gets called prior to displaying the form. Transfer the data
        ' from the input parameters into the display components.

        Public Overloads Overrides Sub TransferDataIn()

            PromptLabel.Text = _prompt

            ' Since the number of radio buttons is variable, we create the list here
            ' and add it into the empty panel in the display.

            _buttons = New List(Of RadioButton)()

            For Each label As String In _labels

                Dim button As New RadioButton()

                _buttons.Add(button)

                button.Text = label

                Dim x As New Drawing.Font("Arial Rounded MT Bold", 8)

```

```

        button.Font = x

        button.Height = 20

        button.Width = 300

        RadioButtonsPanel.Controls.Add(button)

    Next

    If _labels.Length > 0 Then
        If _default <= 0 Then
            _buttons(0).Checked = True

            ElseIf _default >= (_labels.Length - 1) Then
                _buttons(_labels.Length - 1).Checked = True

            Else
                _buttons(_default).Checked = True

            End If

        End If

        Me.CommentPanel.Visible = False

        If _commentVisible Then
            CommentPanel.Visible = True

            CommentText.Text = _comment

            CommentLabel.Text = _commentPrompt

        End If

    End Sub

' This function gets called prior to returning the outputs from the form. Transfer
' data from the display components and set the output fields.

Public Overloads Overrides Sub TransferDataOut()

    For i As Integer = 0 To _labels.Length - 1

        If _buttons(i).Checked = True Then

            _selection = TryCast(_buttons(i).Text, String)

            _selectedIndex = i

        Exit For

    End If

    Next

    If _commentVisible Then

        _comment = CommentText.Text

    End If

End Sub

```

```

''' <summary>
''' This function gets called to validate the form prior to attempting to return parameters.
''' This function gets called before TransferDataOut()
''' </summary>

Public Overloads Overrides Function ValidateForm(ByRef errorMessage As String) As Boolean

    errorMessage = ""

    ' Initialize the output error message

    If _buttons.Count = 0 Then

        errorMessage = "The form is invalid because there are no radio buttons."

        Return False

    End If

    If _commentVisible AndAlso (CommentText.Text.Trim().Length = 0) Then

        errorMessage = "Please enter a comment"

        CommentText.Focus()

        Return False

    End If

    Return True

End Function

#End Region

#Region "InputParameters"

' These are the input parameters for the form. The values of these fields are
' set automatically by the base class when this form is initialized.
' The InputProperty specifies the User Readable Parameter Name and the Help Text for the
' parameter that will be displayed in the UI.

<Global.Proficiency.Platform.Samples.Forms.Base.InputProperty("Prompt", "Prompt", "Prompt for form")> _
Protected _prompt As String = ""

<Global.Proficiency.Platform.Samples.Forms.Base.InputProperty("Labels", "Labels", "Labels for the Buttons")> _
Protected _labels As String() = New String(-1) {}

<Global.Proficiency.Platform.Samples.Forms.Base.InputProperty("Default Index", "Default Index", "Index of Default Radio
Button")> _
Protected _default As Integer = 0

```

```

    <Global.Proficiency.Platform.Samples.Forms.Base.InputProperty("Comment Visible", "Comment Visible", "If true user may
enter a comment, defaults to false")> _
        Protected _commentVisible As Boolean = False

    <Global.Proficiency.Platform.Samples.Forms.Base.InputProperty("Comment Prompt", "Comment Prompt", "Prompt to use for
comment field")> _
        Protected _commentPrompt As String = "Comment"

#End Region

#Region "OutputParameters"

    ' These are the output parameters for the form. The values of these fields are
    ' automatically transferred out after TransferDataOut() is called.

    <Global.Proficiency.Platform.Samples.Forms.Base.OutputProperty("Chosen Label", "Chosen Label", "The Chosen Label")> _
        Protected _selection As String

    <Global.Proficiency.Platform.Samples.Forms.Base.OutputProperty("Chosen Index", "Chosen Index", "The Chosen Index")> _
        Protected _selectedIndex As Integer

    <Global.Proficiency.Platform.Samples.Forms.Base.OutputProperty("Comment", "Comment", "User provided comment")> _
        Protected _comment As String = ""

#End Region

End Class

```

The project must to be compiled to generate the DLL that is used to create the display that will be used to access the form within Workflow.

Reading Data Internally

In addition to input parameters, your form can have input data that is read internally; such as from a data source. This kind of data normally is received from a subscription to a Workflow service provider, and can be displayed in the form.

The code that reads the data should be inserted into the `TransferDataIn()` override, and then the properties of the display objects can be set to values from the data.

An existing sample form (UsersFormVB) shows a list of user names obtained from the Personnel service provider. The code for the override is as follows:

```
Public Overloads Overrides Sub TransferDataIn()

    Try

        Dim userStartup As Proficy.Platform.Core.ClientTools.ServiceDirectory.UserStartup = userStartup.[Default]

        Dim personnelService As Proficy.Platform.Services.Personnel.Interfaces.IPersonnel =
        TryCast(userStartup.GetDefaultProvider("IPersonnel"), Proficy.Platform.Services.Personnel.Interfaces.IPersonnel)

        Dim people As Proficy.Platform.Services.Personnel.DataTypes.Person() = personnelService.GetAllUsers()

        _usersList = New List(Of String)()

        If people IsNot Nothing Then

            For Each person As Proficy.Platform.Services.Personnel.DataTypes.Person In people

                _usersList.Add(person.S95Id)

            Next

            _usersList.Sort()

            UsersListBox.Items.AddRange(_usersList.ToArray())

        End If

    Catch e As Exception

        HeaderLabel.Text = "Error: " + e.Message

        HeaderLabel.ForeColor = System.Drawing.Color.Red

    End Try

End Sub
```

Writing Data Internally

In addition to output parameters being used by Workflow to perform actions, your form can have output data that is written internally; for example, to a log file, or a Proficy service provider.

The code that writes the data should be inserted into the TransferDataOut() override.

An existing sample form (LoggerFormVB) takes a message from the form and writes it to the standard log file. The code for the override is as follows:

```
Private ReadOnly Property Logger() As ILog

    Get

        Return LogManager.GetLogger("LoggerForm", LogContextBehavior.UseThreadLogContext)

    End Get

End Property

Public Overloads Overrides Sub TransferDataOut()
```

```
' There are no output paremeters, but we have this override which is called on Submit.
' It will write the logger data to
' Proficy INstall directory\Proficy Workflow\Program\Logs\ProficyClient.log
' Each line will have [Client.LoggerForm] at the end.
' To see the DEBUG ones you have to lower the logging level.
Dim gsd As New GetStringDelegate(AddressOf SameString)

Select Case LoggerType.SelectedIndex

    Case 0

        Logger.Info(gsd, LoggerText.Text)

        Exit Select

    Case 1

        Logger.[Error](gsd, LoggerText.Text)

        Exit Select

    Case 2

        Logger.Debug(LoggerText.Text)

        Exit Select

    Case Else

        Exit Select

End Select

End Sub

' A simple function returning the same string as was input.

Public Function SameString(ByVal input As String, ByVal culture As Globalization.CultureInfo) As String

    Return input

End Function
```

Related information

[Form Development in Visual Studio \(on page 1224\)](#)

Subscribing to Events

Your form can respond to published events.

The code that subscribes to the event is inserted into the `TransferDataIn()` override, adding your event handler, in which properties of the display objects (or output parameters) can be set to values from the event arguments.

For example, a time event can be set up to recur every 30 seconds, and the form display the time of the most recent event. The code is as follows:

```
Imports System

Imports System.Collections.Generic

Imports System.Windows

Imports Proficy.Platform.Core.Cdo.Connections

Imports Proficy.Platform.Core.ClientTools.ServiceDirectory

Imports Proficy.Platform.Core.ProficySystem.MobileObject

Imports Proficy.Platform.Core.ProficySystem.Types

Imports Proficy.Platform.Samples.Forms.Base

Public Class EventVB

    Inherits BaseFormVB.Proficy.Platform.Samples.Forms.Base.VB.BaseFormVB

    Public Overloads Overrides Sub TransferDataIn()

        ' The published Time Event to subscribe to is coming from Workflow via an input parameter in this case.

        SubscribeToEvent(_directoryResource.LdapAddress )

    End Sub

    Private Sub SubscribeToEvent(ByVal eventLdapAddress As String)

        Dim userStartup As UserStartup = userStartup.[Default]

        Dim serviceProvider As IServiceProvider = New BasicServicesProvider(userStartup.ServiceDirectory)

        Dim subscription As New DirectoryAddressableSubscription(serviceProvider, eventLdapAddress)

        ' Note: DirectoryAddressableSubscription has been renamed to Subscription

        AddHandler subscription.EventOccurred, AddressOf subscription_EventOccurred

        subscription.Subscribe()

    End Sub

    Private Sub subscription_EventOccurred(ByVal sender As Object, ByVal e As EventOccurredEventArgs)

        ' This is called when the event occurs, and the EventArgs may be used to adjust the

        ' values of display objects, or to set output parameter values. In this case we have a

        ' recurring event, and we update a text block with the time of the latest event, changing

        ' color alternately between red and blue.

        Dim newText As String = DateTime.Now.ToString()

        EventTimeText.Text = newText

        If EventTimeText.ForeColor = System.Drawing.Color.Red Then

            EventTimeText.ForeColor = System.Drawing.Color.Blue

        End If

    End Sub

End Class
```

```

Else
    EventTimeText.ForeColor = System.Drawing.Color.Red
End If

EventTimeText.Refresh()

End Sub

[InputProperty("EventToSubscribe", "Event to Subscribe to", "The time event to subscribe to")]
    protected DirectoryResource _directoryResource = null;

End Class

```

This technique can be generalized for any item that can be subscribed to; however, the `EventOccurredEventArgs` will contain different information depending on the type of event. In the above example, this information is ignored, but in general the information can be extracted and displayed in the form.

Common subscribable items are:

- equipment properties
- time events
- condition events
- new work request events
- work request status changed events
- workflow task expired events
- workflow task step expired events

In each case the LDAP address of the item must be located in the `DirectoryViewer`, and then copied and pasted as an argument into the `SubscribeToEvent` call.

Display Equipment Property Changes Example

```

Private Sub subscription1_EventOccurred(ByVal sender As Object, ByVal e As EventOccurredEventArgs)

    Dim message As MobileMessage = e.EventMessage

    Dim dataItemEvent As DataItemChangedEvent = TryCast(message.ObjectInstance, DataItemChangedEvent)

    Dim newValue As ProcessValue = dataItemEvent.NewValue

    SomeLabel.Text = newValue.ToString()

End Sub

```

Display Time Event Data Example

```

Private Sub subscription1_EventOccurred(ByVal sender As Object, ByVal e As EventOccurredEventArgs)

    Dim eventInfo As TimeEvent = TryCast(e.EventData("EventData").GetObject(), TimeEvent)

    SomeLabel.Text = eventInfo.Description

End Sub

```

```
End Sub
```

Related information

[Form Development in Visual Studio \(on page 1224\)](#)

Noting Form State

Your form can note (to some extent) whether the Submit or Cancel button was clicked; such as to set the value of an output parameter.

If Cancel is pressed, neither `ValidateForm` nor `TransferDataOut` are called, so setting an output parameter only has an effect if Submit is pressed. If the parameter is bound to something (for example, an input parameter of a subsequent form), then the target object remains with its default value if Cancel is pressed.

An example of this is contained in the sample form `UsersFormVB`:

```
Public Overloads Overrides Sub TransferDataOut()  
    _selection = UsersListBox.SelectedItem  
    _status = "Form Submitted"  
End Sub
```

Related information

[Form Development in Visual Studio \(on page 1224\)](#)

Form Development in Silverlight

Forms Development in Silverlight

Silverlight is a development platform that allows you to write and run applications for web, desktop, and mobile use. The Silverlight Designer provides the ability to integrate Silverlight forms in the Workflow Task List, Web Task List, and the Global Viewer in Workflow client.



Important:

Support for Silverlight and Silverlight forms will be discontinued with the next major release of the software.

Form Creation

A form cannot be converted from WPF to Silverlight or vice versa after the form creation process starts. During form authoring, be sure to select the correct type of form to create in the Display Editor.

Technical Limitations

As a technical limitation, .NET content can run only in thick clients (that is, the Proficy Client) where the .NET Framework is installed. It cannot run directly in a web browser or directly on a mobile device. Similarly, Silverlight content can run only in a web browser with a Silverlight plug-in installed. Silverlight content can also run on a mobile device that supports Silverlight (such as Windows 7 mobile devices or Symbian Series 60 phones); however, we do not support these devices.

Clients That Can Host the Form

The table below illustrates the form types each client can host:

Form	Proficy Client	Web Task List	Mobile-sized Task List
.NET form	Yes	No	Yes **
Silverlight form	Yes	Yes	Yes

** In general, .NET content cannot run on mobile devices. However, the Mobile-Sized Task List is designed to run on a server and display content on a mobile device using either GraphOn or Terminal Services.

A form that will be run on a mobile device should be designed with the small form factor in mind. However, that same form will work in both a thick client and a web environment.



Note:

Although a Silverlight form can run in all three environments, it is not recommended.

The following topics describe the tasks necessary to build Silverlight forms.

Related information

[Create a Silverlight form for web-based content \(on page 1243\)](#)

[Extend the Silverlight Designer Capabilities \(on page 1246\)](#)

[Form Development in Visual Studio \(on page 1224\)](#)

Create a Silverlight form for web-based content

This procedure is an example of how to create a Silverlight form for use with web-based content.

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. Click **Add Form**.
3. In the **Name** and **Description** fields, enter a name and description for the resource.
4. Click **OK**.
5. In the **Displays** panel, click **Display Editor**.
6. On the **Overview** tab, select the **Use the Designer tab to create web content** option.
7. Click **Save**.
8. Click the **Designer** tab.
9. If this is the first time the fragment has been opened in the **Designer**, enter a name for the project in the **New Project** dialog box.



Note:

Naming conventions for Silverlight projects are the same as those for WPF displays and forms. Project names can contain only letters, numbers, or underscores.

10. Click **OK** to create the project.
11. After adding the required components to the form, click **Save**.

What to do next



Note:

If multiple users are saving Silverlight forms at the same time, the Feedback Zone may display a message indicating that the form failed to build, even though the build process is actually ongoing. In this scenario, the client log file (found at `<installdir>\Proficiency\Logs\ProficiencyClient.log`) includes an entry reading `ERROR: [HttpRequestTimedOutWithoutDetail`. As a result, to save new changes to the form, reload it in one of the following ways.

- In the **Displays** panel, select **Global Viewer**, and then select **Display Editor**

Or

- In the navigator, select another form, and then select the form again.

Add a control to a Silverlight form

Procedure

1. In the Silverlight designer, click the Toolbox tab, and then expand the appropriate control group.
2. Click the control you want to add, and then in the designer, drag an outline where you want the control to appear.

Image Controls in Silverlight Forms

Due to restrictions, the Source property for an image control cannot be set in the XAML. When an image control is used in the Silverlight designer, the URI for the image source must be manually entered into the image control's Source property. The designer automatically configures the Source property in code behind.

If the form will be edited later in Visual Studio and the Source property will be set from Visual Studio, it must be set in code behind, not in XAML. This can be done by adding code, such as the following, to the InitializeImageSources method in MainPage.Proficy.cs:

```
Image0.Source = new BitmapImage(new Uri(uriString, UriKind.Absolute));
```

Image sources must come from a Web server (not a file:// style URL). The Web server must have a clientaccesspolicy.dita in its root directory. Thus, if the URL for your image is `http://mywebserver:8000/images/animals/kitty.jpg`, the client access policy must have a URL of `http://mywebserver:8000/clientaccesspolicy.dita`.

Calendar controls in Silverlight Forms

The Calendar control's Selected Date property displays the current date in the control. The DisplayDate property represents the date that will be displayed. If this is not set, then the Selected Date is displayed.

To access the Selected Date property in the Binding dialog box or the Expression Editor, clear the Hide Advanced Properties check box.

To bind to the Selected Date or to use it in an expression, bind directly to Selected Date; do not use the Value sub-property.

To work with subproperties of SelectedDate, you must modify the XAML (for bindings) or the MainPage.Proficy.cs (for expressions) files by hand. Editing the MainPage.Proficy.cs file must be done in Visual Studio.



Note:

For more information, see *Edit forms using Visual Studio 2010*.

One method of binding to subproperties by modifying the XAML by hand is to bind to the desired subproperty under the Value subproperty in the designer, and then in Visual Studio, edit the affected file and remove the ".Value" and/or "__Value" from the binding. The binding has one ".Value" to remove. The expression has one ".Value" and one "__Value" to remove.

The following table lists the calendar control bindings that can be used and the bindings that will not work.

Bindings that work	Bindings that do not work
Calendar1.SelectedDate	Calendar1_SelectedDate.Value
Calendar1_SelectedDate.DayOfYear	Calendar1_SelectedDate.Value.DayOfYear

Expressions that work	Expressions that do not work
{Calendar1__SelectedDate__Value__DayOfYear}	{Calendar1__SelectedDate__Value}
{Calendar1__SelectedDate__DayOfYear}	{Calendar1__SelectedDate}

Edit code behind

This procedure provides an example of how to edit the code behind in a Silverlight form.

Before you begin

Procedure

1. In the Project tab, expand MainPage.xaml to reveal MainPage.xaml.cs.

The code behind editor does not yet support IntelliSense, so adding code must be done carefully. To add a pop-up message box from code behind, modify the MainPage.xaml.cs file as follows:

Register an event handler in the constructor (adding your code just after InitializeProficyData(), as shown):

```
public MainPage()
{
    InitializeProficyData();
    Button1.Click += SayHello;
}
```

Add the handler code to MainPage.xaml.cs:

```
private void SayHello(Object sender, EventArgs args)
{
    MessageBox.Show("Hello");
}
```

2. Click **Save** to save the project and its output.
3. Select the form in the Workflow client and view it in the Global Viewer to see the compiled form running.

Silverlight Form Restrictions

There are several restrictions in place for the design of Silverlight forms.

- Only one user should edit a form at a time in the Silverlight Designer. If two users edit at the same time, unexpected errors could occur. The last user who saves can overwrite the changes of the other user.
- Updating a form while others are requesting the form may result in an error. Form XAP files are cached in the WebModulesCache directory. For example, if user A is downloading the old form in the Web Task List, while user B is updating the form in the Silverlight Designer, and user C is requesting the new form in the Web Task List, an error could occur.

Extend the Silverlight Designer Capabilities

The Silverlight Designer can be extended so that user-created controls or those from a third-party control vendor are available in the Silverlight Designer toolbox.

Code that was written in a separate Silverlight class library can also be invoked from code behind. The scenarios are outlined in the following topics:

Add a custom control to the Silverlight Designer

The controls that appear in the Silverlight Designer toolbox are Silverlight controls. To add additional controls into the Designer toolbox, start with a Silverlight control from a commercial control vendor or create your own.

Procedure

- To use an existing Silverlight control, follow the instructions in *Add a reference to a Silverlight assembly*.
- To create your own control, create it in Visual Studio as a Silverlight class library project. Test it within Visual Studio to validate that the control works as it should before trying to use it in the Silverlight Designer. Then, follow the instructions in *Add a reference to a Silverlight assembly*.

**Note:**

For either procedure, see *Create a custom data type editor for the Silverlight Designer* to customize the design-time experience with the control.

Related information

[Add a reference to a Silverlight assembly \(on page 1254\)](#)

[Create a custom data type editor for the Silverlight Designer \(on page 1250\)](#)

Restrictions and Tips for Creating a Silverlight Control

When creating a custom control for Silverlight, follow the tips below for standards of use.

- The control must derive from `System.Windows.FrameworkElement`.
- The control must have a parameterless constructor.
- Any property that you want to be able to configure through the Designer should be exposed as a standard Silverlight dependency property.
 - The property can be driven by a binding or expression in the Silverlight Designer.
 - For simple data types, the Designer provides a default property editor to all users so that a literal value for the property can be configured.
 - For data types where the default property editor is not sufficient, custom data type editors can be created. For more information, see *Create a Custom Data Type Editor for the Silverlight Designer*.
 - These properties should be of a data type that can be serialized to XAML.

**Note:**

There is a known issue with using GUID as a data type.

- For complex data types, if a property using this type is not being persisted, it may help to initialize the default value of the property using the `SetValue` method as described below.

- If you want the property to be configurable through the designer, the Editor-Browsable attribute should be set to Always or Advanced.
- The Category attribute can be used to specify the category this property appears in.

Below is an example showing the instructions property to be exposed on My CustomControl. For the definition below, this property appears in the Designer with a default value of "Please enter in the values below." A different value for this property can be specified through the default property editor provided. In this case, it is a simple textbox. In addition, this property value can be driven by a binding or expression in the Silverlight Designer.

```

public static readonly DependencyProperty InstructionsProperty
=
DependencyProperty.Register("Instructions", typeof(string),
typeof(MyCustomControl), new
PropertyMetadata("Please enter in the values below.));
    [EditorBrowsable(EditorBrowsableState.Always),
    Category("Definition")]
    public String Instructions
    {
        get { return GetValue(InstructionsProperty) as
String; }
        set { SetValue(InstructionsProperty, value); }
    }

```

For complex data types, a default value may need to be specified inside the control constructor instead of the DependencyProperty definition. It is required only if the custom property is not being serialized to XAML when the property is configured from within the Silverlight Designer. See the example below:

```

public HelpfulDataGrid()
    {
        SetValue(PopupDataProperty, new CustomList());
        //Rest of constructor code here
    }

```

After the control is created and tested from within Visual Studio, refer to *Add a reference to a Silverlight assembly*, which explains how to configure the control so it appears in the toolbox.

Related information

[Add a reference to a Silverlight assembly \(on page 1254\)](#)

[Create a custom data type editor for the Silverlight Designer \(on page 1250\)](#)

Design Mode

It may be helpful to give the custom control different design-time behavior than run-time behavior.

First, determine if the control is on a design surface in design mode.

A static class method, `DesignModeCheck.IsInDesignMode`, is available to use as shown below. It works in both the Visual Studio designer and the Proficy Visualization designer environments.

```
Loaded += (s, e) =>
{
    if (DesignModeCheck.IsInDesignMode(this))
    {
        LayoutRoot.Visibility = Visibility.Visible;
        _titleTextBlock.Text = this.GetType().Name;
    }
    else
    {
        LayoutRoot.Visibility = Visibility.Collapsed;
    }
};
```

Related information

[Restrictions and Tips for Creating a Silverlight Control \(on page 1247\)](#)

Custom Data Type Editors from Command Lines

Custom data types and their property editors can be configured with two command line utilities: `UpdateSilverlightAssembly` and `UpdateSilverlightDataType`. These utilities are located in the program directory.

Create solutions to ship on top of Workflow

This procedure describes a scenario about a team that wants to provision a system they ship by installing forms, displays, toolbox assemblies, and custom data editors at install time.

Procedure

1. Add Silverlight assemblies containing the custom data types and/or editors, but use an input file with the assembly file names specified:

```
UpdateSilverlightAssembly -u username -p password -a
CustomDataTypeAssemblies.txt
```

2. Create a comma-separated file listing the custom data types, custom data type editors, and their assemblies. The following sample file can be created with the `UpdateSilverlightDataType -s` command line:

```
*DataType assembly filename,DataType class name,DataType editor assembly filename,DataType editor class name
Proficy.Visualization.SampleControls.dll,SampleControls.CustomList,Proficy.Visualization.SampleControls.Design.
dll,SampleControls.Design.CustomListPropertyEditor
Proficy.Visualization.SampleControls.dll,SampleControls.CustomItem,Proficy.Visualization.SampleControls.dll,Sam
pleControls.Design.CustomItemPropertyEditor
```

3. Add custom data types and their property editors with the `UpdateSilverlightDataType` command. If the assemblies in the input CSV file do not have a full path, then put it in the same directory as the assemblies.

```
UpdateSilverlightAssembly -u username -p password -a Input.csv Output.csv
```

Custom Data Type Editors

The following topics explain how to create and configure a custom data type editor.

Data Type Editor Limitations

A custom data type editor is a Silverlight assembly. It has the same security restrictions as any other Silverlight assembly.

Be aware of the following limitations:

- You do not have access to local storage.
- You are subject to the network access restrictions of Silverlight.
- You must use the Proficy proxy generator to make calls to a Proficy service provider.

Create a custom data type editor for the Silverlight Designer

A custom data type editor can be created for use in the Silverlight Designer for data types where no default property editor is provided.

About this task**Important:**

Any custom data type editors that were written to work with Visual Studio or Expression Blend will not work in the Silverlight Designer because it is a Silverlight application that requires all loaded code to be in Silverlight. Visual Studio and Expression Blend are desktop applications, so data type editors written for them are not written in Silverlight.

Procedure

1. Create the data type editor in Visual Studio as a Silverlight 4 class library project. The data type editor can be in a separate class library from the control definition.

**Note:**

The data type editor will be used only by the Silverlight Designer at design time and not when the control is being run from the Workflow client or Web Task List at run time.

2. Add the following references to this project.

The data type editor should derive from

```
Proficy.Platform.Core.DisplayFramework.Controls.Design.PropertyEditorBase
```

- Proficy.Platform.Core.DisplayFramework.dll
- Proficy.Platform.Core.DisplayFramework.Design.dll
- AuroraCommon.dll
- AuroraInterfaces.dll

**Note:**

- For 32-bit systems, the references reside in the <Installed directory>:
 \Program Files\Proficy\Proficy Workflow\Silverlight directory.
- For 64-bit systems, the references reside in C:\Program Files
 (x86)\Proficy\Proficy Workflow\Silverlight directory.

3. Two methods are required for implementation:

```
/// <summary>
/// Returns the UI for this property editor. This method will only be called once.
/// This UI is responsible for detecting when the property Value has changed, and setting
/// PropertyValue with the new value when this occurs. For example, is this editor launches
/// a dialog to configure the property, PropertyValue can be set when the dialog closes.
```

```

/// </summary>
public override FrameworkElement GetPropertyEditorUI()

/// <summary>
/// This method is called when PropertyValue changes anytime after initialization is complete.
/// This includes changes initiated by the user and when Reset to Default is selected. Any updates
/// to the property editor UI that would need to occur in these cases should be added here.
/// The PropertyValue can be used to access the new value of the property configured by this
/// editor. GetPropertyEditorUI will be called prior to this method being called.
/// </summary>
protected override void OnValueChanged()

```

Other helper methods available from PropertyEditorBase that call from GetPropertyEditorUI include:

```

/// <summary>
/// Returns a control that can be used to show a read-only view of the item.
/// </summary>
/// <param name="text">The text to show.</param>
/// <param name="toolTip">The tooltip, if any, to be added for this control.</param>
protected Control CreateReadOnlyControl(string text, string toolTip)

```

```

/// <summary>
/// Returns a default browse button.
/// </summary>
protected Button CreateBrowseButton()

```

```

/// <summary>
/// Forces all the properties currently displayed in the Property panel to be re-evaluated. This
/// method should be used if setting the current property might affect the values of other
/// properties.
/// </summary>
protected void ForceRefreshOfProperties()

```

4. Configure the custom data type editor to be used in the Silverlight Designer.

Configure a custom data type editor for the Silverlight Designer

The data type and data type editor should be configured so the Silverlight Designer can use the editor to edit the data type.

Before you begin**Note:**

Any configuration made here applies to all Silverlight Designer forms, not just the one that is currently selected in the navigator.

Procedure

1. From the navigator, click **Global Displays > Forms**.
2. From the Forms list, select the form you want to work with.
3. In Show Compatible panel, select **Display Editor**.
4. Click the **Silverlight Config** tab.
5. Under Toolbox Controls, click **Add** to add the custom data type editor assembly, if it is not already listed.
6. Click the **Data Type Editors** tab.
7. In the Data Types area, click **Add**.
8. For the Custom Data Type, click **Set** to choose the assembly.
9. Browse to the .dll file that contains your data type editor, and then click **Open** to select it.
10. From the **Type** drop-down list, select the custom data you want to use.
11. In the Editor area, click **Set**.
12. Browse to the .dll file that contains your data type, and then click **Open**.
13. From the **Type** drop-down list, select the editor class you want to use.

**Note:**

If the custom data type or the data type editor are not in assemblies that have already been added as references, follow the instructions in *Add a reference to a Silverlight assembly*. This makes the data type editor available to the Silverlight Designer.

14. Click **Save**.

Related information

[Add a reference to a Silverlight assembly \(on page 1254\)](#)

Configure a custom data type editor for a dependency property

A custom data type editor can be created and configured to use a dependency property.

Before you begin

To ensure that changes from this editor are persisted in XAML:

- Add a reference to WebApplication.dll
- Use the extension method, `SetLocalValue`, to force the value to be written to XAML

For example, in a custom control with a property of type `TestProperty`, `TestProperty` is the custom data type and it is a `DependencyObject`. It has a `DependencyProperty`, `ValueAsStringProperty`.

Procedure

1. In the custom editor for `TestProperty`, assign the custom data type property value (`this.PropertyValue`) to set the value for `TestProperty`.
2. In addition, since the custom data type has a dependency property, call `SetLocalValue` so that its value is written to XAML.

```
string newTextValue = _ui.Text;

TestProperty tp = new TestProperty(newTextValue);

this.PropertyValue = tp;

tp.SetLocalValue(TestProperty.ValueAsStringProperty, _ui.Text);
```

Call into a class library

To invoke code that was written in a separate Silverlight class library, a call must be placed to that class library.

The call must be invoked first, and then the library can be leveraged from the code behind in the Silverlight project. To edit code from Visual Studio rather than the Silverlight Designer, see *Edit Forms using Visual Studio 2010*.



Note:

To load the assembly so you can use the classes in it, follow the instructions in *Add a reference to a Silverlight assembly*.

Related information

[Add a reference to a Silverlight assembly \(on page 1254\)](#)

[Edit forms using Visual Studio 2010 \(on page 1263\)](#)

Add a reference to a Silverlight assembly

To add new controls to the Silverlight Designer toolbox or to call code in a separate assembly, specify the containing assemblies from the **Display Editor**.

Before you begin



Note:

Files added here will be available to all Silverlight forms, and not just the one currently selected in the Navigator.

Procedure

1. In the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the form you want to work with.
3. In the **Displays** panel, click **Display Editor**.
4. Click the **Silverlight Config** tab.
5. Click **Add**.
6. Select the Silverlight assembly that will be referenced. Each of the assemblies specified here is added as a reference to all Silverlight projects. Currently, a reference cannot be added to only a specific project.
7. Click **Save**.



Note:

The next time the Silverlight Designer is opened, the reference will be available for use from any Silverlight project. Any custom controls defined in any of the assemblies will appear in the Silverlight Designer toolbox under the **Custom Controls** category.

Update a Silverlight reference

To update a reference that was written in a separate Silverlight class library, a call must be placed to that class library.

Procedure

1. In the Display Editor, on the Silverlight Config tab, select the reference you want to remove, and then click **Remove**.
2. Click **Add** to re-add the reference, and then click **Save**. Recompiling the file to disk is not sufficient since this reference must be saved to the server.



Note:

The referenced assembly is bundled into the form's XAP file. Any projects built against a specific referenced assembly version continue to use that older version, even if the reference is updated in the Silverlight Config interface, because the older version is referenced in the form's XAP file.

3. To use the new version of the assembly, open the form's project in the Silverlight Designer, and then save and build it.

A new XAP file is created.

**Note:**

If changes to the assembly include removal of controls or modification of public properties, a screen may appear that indicates there was a load failure.

The error in the Feedback zone attempts to indicate what the issue is. Often, the XAML file must be manually edited to correct the load failure. For example, the following line defining the Silverlight Control should be removed because the reference to SilverlightClassLibrary1.dll where this control was defined was removed:

```
<scl:SilverlightControl .../>
```

4. Click **Save** after removing the line.
5. Click **Open**, and then reload the project to see it load properly.

Related information

[Edit forms using Visual Studio 2010 \(on page 1263\)](#)

Add custom submit, cancel, and enable to a form using the Designer

The form can be configured so that the Workflow Task List Submit and Cancel buttons are not visible when the form is displayed in the Task List. You can also ensure that a form is only ready for submission after the appropriate operator interaction by enabling/disabling the **Submit** button.

Procedure

1. From the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. In the **Displays** panel, click **Display Editor**.
4. In the **Document Tree** tab, select the root of the document tree.
5. In the root element's property panel, clear the **IsSubmitVisible** and **IsCancelVisible** check boxes to hide the Task List Submit and Cancel buttons.
6. **Optional:** To wait for operator interaction before enabling the **Submit** button, from the code behind, set the **IsSubmitEnabled** property to false to disable the **Submit** button.
7. To add code-designed forms to a workflow, see *Configure a Form activity*.

Add custom auto submit and auto cancel to a Silverlight form using Visual Studio

If a form is created in Visual Studio rather than the Designer, custom submit and cancel buttons can be configured.

Before you begin

Procedure

1. Open the form in Visual Studio.
2. Browse to the Silverlight folder of the Proficy Workflow installation directory, and then open the MainForm.xaml file.
3. In the References section of the project, add a reference to DisplayFramework.
4. For each Tasks List button that will be hidden, add the appropriate visibility-attached property to the form.

```
<UserControl
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Name="RootElement"
xmlns:vpic="clr-namespace:VantagePoint.InputControls;assembly=StatusControls"
xmlns:ppcdf="clr-
namespace:Proficy.Platform.Core.DisplayFramework.Forms;assembly=Proficy.Platform.Core.DisplayFramework"
ppcdf:FormProperties.IsSubmitVisible="False"
ppcdf:FormProperties.IsCancelVisible="False"
x:Class="SampleForm_VSSL.MainPage">
<UserControl.Content>
```

5. To add code-designed forms to a workflow, see *Configure a Form activity*.

Configure custom submit, cancel, and/or enable behavior in Silverlight forms

Submitting or cancelling a form is determined during the design of the workflow containing the form.

Procedure

1. From the **Project** tree, open the file, MainPage.xaml.cs.
2. In the **Main Page** constructor, register the handlers with the button click events by using the code below:

```
public MainPage()
{
```

```

InitializeComponent();

_interfaceHelpers[typeof(IResourceParameters)] = new FragmentMetadataByAttributes(this);
_interfaceHelpers[typeof(IFragmentInitialization)] = new FragmentInitialization(this);

InitializeProficyData();

MySubmitButton.Click += OnSubmit;
MyCancelButton.Click += OnCancel;
}

```

3. In each handler, add any custom code that is required, followed by the code necessary to submit or cancel and close the form.

```

public void OnSubmit(object sender, EventArgs args)
{
    //Custom Code
    MessageBox.Show("Submitting");

    //Code required to submit the form
    if (FormHost != null)
    {
        FormHost.SubmitCommand.Execute();
    }
}

public void OnCancel(object sender, EventArgs args)
{
    //Custom Code
    MessageBox.Show("Cancelling");

    //Code required to cancel the form
    if (FormHost != null)
    {
        FormHost.CancelCommand.Execute();
    }
}

```

4. The **Submit** button can be enabled or disabled depending on the interaction of the operator with the Tasks List. For example, by adding the following call method and setting the value in

the form code, the **Submit** button becomes disabled after the operator performs an action:

```
FormProperties.SetIsSubmitEnabled(this, false).
```

You can use this value to ensure that a form is submitted only after valid data is entered to prevent premature form submission.

5. To add code-designed forms to a workflow, see *Configure a Form activity*.

Configure visibility of submit/cancel buttons for Silverlight forms hosted in the Task List

To create custom submit and cancel buttons in your Silverlight `SilverlightStringObject` form, you can configure the **FormProperties** class in order to set the visibility of the default buttons to false.

Procedure

1. From the **Project** tree, open the file, `MainPage.xaml.cs`.
2. In the **Main Page** constructor, set the form property values by using the code below.

```
public MainPage()
{
    ProficiencyComposition.SatisfyImports(this);
    InitializeComponent();
    _interfaceHelpers[typeof(IResourceParameters)] = new FragmentMetadataByAttributes(this);
    _interfaceHelpers[typeof(IFragmentInitialization)] = new FragmentInitialization(this);

    InitializeProficiencyData();

    FormProperties.SetIsSubmitVisible(this, false);
    FormProperties.SetIsCancelVisible(this, false);
}
```

3. The **Submit** button can be enabled or disabled depending on the interaction of the operator with the Task List. For example, by adding the following call method and setting the value in the form code, the **Submit** button becomes disabled after the operator performs an action:

```
FormProperties.SetIsSubmitEnabled(this, false).
```

You can use this value to ensure that a form is submitted only after valid data is entered to prevent premature form submission.

4. To add code-designed forms to a workflow, see *Configure a Form activity*.

Configure visibility of the scroll bar for Silverlight forms hosted in the Task List

To disable the horizontal and vertical scroll bars in forms used in a web-based Task List, you can configure the **FormProperties** class in order to set the visibility to false.

Procedure

1. From the **Project** tree, open the file, `MainPage.xaml.cs`.
2. In the **Main Page** constructor, disable the **ScrollBar** form property values by using the code below.

According to the Microsoft `ScrollBarVisibility` enumeration in the `System.Windows.Controls` namespace, the possible values for scroll bar visibility are:

- `ScrollBarVisibility.Auto`
- `ScrollBarVisibility.Disabled`
- `ScrollBarVisibility.Hidden`
- `ScrollBarVisibility.Visible`

```
public MainPage()
{
    ProficiencyComposition.SatisfyImports(this);
    InitializeComponent();
    _interfaceHelpers[typeof(IResourceParameters)] = new FragmentMetadataByAttributes(this);
    _interfaceHelpers[typeof(IFragmentInitialization)] = new FragmentInitialization(this);

    InitializeProficiencyData();

    // Example: Setting the Web Task List's ScrollBar Visibility property to Disabled
    FormProperties.SetHorizontalScrollBarVisibility(this, ScrollBarVisibility.Disabled);
    FormProperties.SetVerticalScrollBarVisibility(this, ScrollBarVisibility.Disabled);
}
```

3. To add code-designed forms to a workflow, see *Configure a Form activity*.

Special Considerations for Custom Cancel in Silverlight Forms

The ability to cancel a form is determined during the design of the workflow containing the form.

Unlike the Task List Cancel button, there is no mechanism in place to disable a custom cancel button when the form is configured to disallow canceling.

If you click a custom cancel button on a form where canceling is not permitted, the cancel operation fails with an exception that is noted in the feedback zone.

Build custom data types

The following procedure describes a scenario where a developer wants to use commands in a Post-Build step in a Visual Studio project to update the forms, displays, toolbox assemblies, and custom data editors.

Before you begin

Procedure

1. Add Silverlight assemblies containing the custom data types and/or editors with the

`UpdateSilverlightAssembly` command.

```
UpdateSilverlightAssembly -u username -p password -a MyCustomDataTypes.dll
UpdateSilverlightAssembly -u username -p password -a MyCustomDatatypeEditors.dll
```

2. Create a comma-separated file listing the custom data types, custom data type editors, and their assemblies. Here's a sample file, which can be created with the `UpdateSilverlightDataType -s` command line:

```
*DataType assembly filename,DataType class name,DataType editor assembly filename,DataType editor class name
Proficy.Visualization.SampleControls.dll,SampleControls.CustomList,Proficy.Visualization.SampleControls.Design.
dll,SampleControls.Design.CustomListPropertyEditor
Proficy.Visualization.SampleControls.dll,SampleControls.CustomItem,Proficy.Visualization.SampleControls.dll,Sam
pleControls.Design.CustomItemPropertyEditor
```

3. Add custom data types and their property editors with the `UpdateSilverlightDataType` command. If the assemblies in the input csv file do not have a full path, then put it in the same directory as the assemblies.

```
UpdateSilverlightAssembly -u username -p password -a Input.csv Output.csv
```

Specify Parameter Metadata for Silverlight Forms

When parameter metadata is extracted from a form, the following attributes are parsed:

Attributes	Description
System.ComponentModel.EditorBrowsable(EditorBrowsable)	Indicates that the Silverlight property should be exposed as a parameter

Attributes	Description
System.ComponentModel.DescriptionAttribute	Provides a description of the parameter that is displayed in the Workflow editor
ParameterRenameAttribute	Allows the parameter to have a different name than the Silverlight property
ParameterDisplayNameAttribute	Changes the name of the parameter as presented to the user
InputAttribute	Indicates that the property will be treated as an input to the form. If the property does not have a public set method then an error will be generated while extracting the metadata
OutputAttribute	Indicates that the property will be treated as an output from the form. If the property does not have a public get method then an error will be generated while extracting the metadata
ParameterDirectoryResourceMetadataAttribute	Provides additional metadata for properties that have a type of DirectoryResource
DefaultValue	Provides a default value for the parameter. The argument to the attribute is a string. This string is then converted to the data type of the parameter. If the conversion fails then an exception is thrown

The InputAttribute and OutputAttribute attributes should be used to indicate if the property should be treated as an input or an output parameter. If neither is present then a property with a public set method will be treated as an input parameter. If there is no public set method and there is a public get method then the property will be treated as an output parameter. Examples of properties with attributes are shown below:

- ```

[EditorBrowsable]

[ParameterDisplayNameAttribute("Property With Input And Output")]

[Input]

[System.ComponentModel.DefaultValueAttribute("default value")]

[System.ComponentModel.Description("PropertyWithInputAndOutput Description")]

public string PropertyWithInputAndOutput { get; set; }

```
- ```

[EditorBrowsable]

[System.ComponentModel.Description("PropertyWithInput Description")]

[ParameterRenameAttribute("PropertyRenamed")]

public int PropertyWithInput { set { } }

```
- ```

[EditorBrowsable]

[Output]

[System.ComponentModel.Description("PropertyWithOutput Description")]

public string PropertyWithOutput { get { return ""; } }

```
- ```

[EditorBrowsable]

[System.ComponentModel.Description("DirectoryResourceOutput Description")]

[ParameterDirectoryResourceMetadata("adq", "1,2", "3,4")]

public DirectoryResource DirectoryResourceOutput { get { return new
DirectoryResource(""); } }

```

Edit forms using Visual Studio 2010

Take care when editing Designer-created forms in Visual Studio. It is possible to make changes that prevent the Designer from opening the project. The following notes will be helpful in creating changes that are compatible with the Designer.

Limitations

Avoid editing the following files:

- App.xaml
- App.xaml.cs
- DataItems.dat
- MainPage.Proficy.cs

The following guidelines are also important:

- Do not change the project name.
- Do not change the project namespace.
- Do not change the XAP file name.
- Do not set the Source property for Image controls in XAML. (You will get errors later if you try to open the XAML source in the Proficy Designer). Instead, set the Source property in code behind (for example, `_myImage.Source = new BitmapImage(imageUrl)`).

Add new files

New files can be added to the project using either the Silverlight Designer or Visual Studio. However, files added using Visual Studio will not be listed in the Silverlight Designer Project tab.

Locate the project files

Locate the directory where the projects are stored. The location is under the directory where Proficy is installed. The default install directory for Windows Server 2003 is `C:\Program Files\Proficy\Proficy Workflow\Program\Designer\Projects`. The project files are stored in a sub-directory with the same name as the project.

Allow remote access to the project files

An administrator of the server can allow remote access to the project files by creating a share. The share can be for either `C:\Program Files\Proficy \Proficy Workflow\Program\Designer\Projects` or for individual project directories.

Related information

[Specify Parameter Metadata for Silverlight Forms \(on page 1261\)](#)

[Debug Content Created in the Silverlight Designer \(on page 1266\)](#)

Edit the form

The project is stored in a database on the server. The project must be copied to the filesystem before it can be edited using Visual Studio.

Procedure

1. From the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. In the **Displays** panel, click **Display Editor**.
4. Click the **Designer** tab.

The project for the form will now be located on the server.

**Note:**

If this is the first time the project has been opened, the project must be saved and assigned a project name. Save the form to store the project in the database as well as in the filesystem.

5. Note the name of the project, which may be different than the name of the form. The name of the project is the root node in the Project tab of the Silverlight Designer.
6. Close the Proficy Client or make sure the form is not open in the Display Editor, either by navigating to a different form or selecting a different view in the Show Compatible panel.
7. Open the project in Visual Studio 2010.

**Note:**

Sometimes when a project is opened in Visual Studio using a share to the server, it may not compile or run. This is caused by a security issue in Visual Studio.

If problems are encountered when compiling or running the file, try the following:

- Close Visual Studio.
- Open the file `devenv.exe.config` for editing. This file can be found in `C:\Program Files\Microsoft Visual Studio 10.0\Common7\IDE`.
- Locate the section `<runtime>`.
- Add the following entry: `<loadFromRemoteSources enabled="true" />`.
- Save the file.
- Re-open the project.

8. Edit the form project.
9. Save the project, and then exit Visual Studio.

Related information

[Edit forms using Visual Studio 2010 \(on page 1263\)](#)

Save the form

After editing is complete, update the form in the database.

Before you begin

Procedure

1. From the navigator, click **Global Displays > Forms**.
2. From the **Forms** list, select the resource you want to work with.
3. In the **Displays** panel, click **Display Editor**.
4. Click the **Designer** tab.
5. Click **Save**.



Note:

You may need to make minor changes to the form to enable the Save button. The form will be rebuilt and saved to the database.

Related information

[Debug Content Created in the Silverlight Designer \(on page 1266\)](#)

Debug Content Created in the Silverlight Designer

Content created in Silverlight can encounter bugs. This topic provides assistance in debugging your project.

There are three methods to debug the form project:

- Run the form in Visual Studio 2010.
- Attach to the Web Task List process.
- Attach to the Workflow client process, using the Task List or Global Viewer to view the form.

Related information

[Use logging and FeedbackService \(on page 1269\)](#)

Run the form in Visual Studio 2010

Running the form in Visual Studio can help identify bugs in content.

Before you begin

Procedure

1. Open the form in Visual Studio.
2. Initialize any input parameters in the form. If the form has any input parameters, they must be initialized.
 - a. Open **MainPage.xaml.cs**.
 - b. Locate the constructor for the MainPage class.
 - c. Assign values to the input parameters.
3. Run the project.
4. If errors are reported about not being able to connect to the server during debugging, you must debug it using one of the other methods.

Related information

[Edit forms using Visual Studio 2010 \(on page 1263\)](#)

Attach to the Web Task List or Workflow Client

If you cannot run the form independently, you can debug while running the form in the Web Task List or Workflow Client.

This procedure involves:

- adding the form to a workflow
- preparing to debug the form
- activating the form

Add the form to a workflow

Before you begin

Procedure

1. From the Navigator, select the **Workflow > Workflows**, and then add a new workflow.
2. From the Show Compatible panel, select **Workflow Editor**.
3. On the Overview tab, set the status to **Enable**.
4. On the Workflow Editor tab, add a local subprocess.
5. Set the Task Step Assignments to **Override Workflow Task Assignments**.
6. Set the Personnel field to the user that will log in to the Web Task List.
7. Add a Form activity to the subprocess.

8. In the Form field, click the ellipsis button to select the form you want to debug.
9. Click **Save**.

Related information

[Edit forms using Visual Studio 2010 \(on page 1263\)](#)

Prepare to debug the form

Procedure

1. Locate the project on the server. The form's project can be found in the <Install Directory> \Proficy Workflow\Program\Designer\Projects\{Project Name} directory on the server.
2. Open the project in Visual Studio 2010. It will have the name **<Project Name>.csproj**.
3. Set breakpoints.
4. Attach the project to the process to be debugged.
5. To debug:
 - In the Web Tasks List:
 - a. Close all Internet Explorer instances. This will make it easier to identify the Internet Explorer instance that is hosting the Web Task List.
 - b. Start the Web Task List, but do not log in.
 - c. Attach to the Internet Explorer process that is hosting the Web Task List. (In Visual Studio 2010, go to the Debug menu and bring up the Attach to Process dialog. Look for the Internet Explorer processes. Look for the instance that has a type of Silverlight. Attach to that process.)
 - Inside of the Workflow client:
 - a. Attach to the ProficyClient.exe process, with the **Attach to** field in that dialog box set to **Silverlight code**.

Related information

[Edit forms using Visual Studio 2010 \(on page 1263\)](#)

Activate the form

Procedure

1. Start the Proficy Client.
2. In the Navigator, click **Workflow > Workflows**.
3. In the Workflows list, select the workflow that you created previously.

4. In the Workflow Editor, select the **Test/Debug** tab.
5. Click the **Test** button.
6. If you are debugging in the Web Task List, log in to the Web Task List. If you are debugging in the Proficy Client, click the **View My Task List** button in the title bar. The form loads and the break points will be active.

Related information

[Use logging and FeedbackService \(on page 1269\)](#)

Use logging and FeedbackService

In Designer-built forms, you can write messages to both the Workflow client log file and the feedback zone. This requires editing the code behind in the MainPage.xaml.

For example, insert the highlighted code snippet into your MainPage constructor:

```
public MainPage()
{
    Loaded += (o, e) =>
    {
        var message = String.Format("Loaded form {0}",
            GetType().FullName);
        FeedbackService.AddMessage("LoadUnload", FeedbackMessageType.Info,
            message);
        _logger.Info(message);
    };
}
```

A more common use of the logger is to catch various exceptions and log them:

```
void TrySomethingRisky()
{
    try
    {
        ...
    }
    catch (Exception e)
```

```
{  
    _logger.Error(e.Message);  
}  
}
```

Related information

[Debug Content Created in the Silverlight Designer \(on page 1266\)](#)

[Form Editing and Debugging \(on page 1270\)](#)

Form Editing and Debugging

When you are testing a workflow that contains a form activity, the Workflow Debugger can be used to view the inputs and outputs of the form. However, there are times when a greater level of information is desired to understand what is happening inside the form itself.

This section discusses how to debug the code inside a form to get a better understanding of why the form is behaving as it is.

There are two different scenarios described in this section. One is debugging a custom form that was created using Visual Studio. The other is debugging a form that was created with the Forms Designer, and also describes the capability to edit that form in Visual Studio, if necessary.



Important:

You must have Visual Studio 2008 installed on the machine on which you want to debug for either of the scenarios in the following sections to work.

Edit and debug content created in Visual Studio

This scenario deals with cases where you created a form or user display using Visual Studio, and then uploaded this form in the Display Editor of the Proficy Client.

Before you begin

Procedure

1. Locate the required files.

In order to debug your form, the following files are required:

- The source code for your forms.

In this scenario, the source code should be found somewhere on your system; it is not persisted in the Workflow database.

- The symbol files for the form.

There is one symbol file for each project you created, and symbol files are needed only for the assemblies that you need to debug into.



Tip:

Symbol files match the file name, but have a .pdb extension instead of .dll, and are found in the same folder where the .dll is built to. As the .dll is rebuilt, so is the .pdb, so these files must match each other in order to debug properly.

2. Make the .PDB files available to the Client for debugging.

This can be done in one of two ways:

- Copy the .pdb files that correspond to the form assembly and dependencies to the My Documents\Proficy Workflow\UI Assemblies\ folder on the machine on which you want to debug the form.
 - When you are finished debugging the form and are ready to run in production again, delete these files from the My Documents\Proficy Workflow\UI Assemblies\ folder; otherwise, the symbol information will continue to be loaded each time the form is loaded.
- Add the .pdb files as additional files in the **Display Editor** in the Workflow client. This automatically copies the file to the My Documents\Proficy Workflow\UI Assemblies\ folder for all machines when they try to load this form.
 - When you are finished debugging the form and are ready to run in production again, remove this as an additional file in the **Display Editor**. Be aware that this file will remain on any client machines that have already downloaded it, so you must also manually delete the file from the My Documents\Proficy Workflow\UI Assemblies\ folder. Otherwise, the symbol information will continue to be loaded each time the form is loaded.

3. Open the project code using Visual Studio.

Open the code files you want to debug into and set your breakpoints by clicking in the extreme left margin. Putting a breakpoint in the constructor of your form is a good idea in order to test a breakpoint being hit.

4. Launch the Workflow client if it is not already running.
5. In the **Debug** menu of Visual Studio, select **Attach to Process**, and then select **ProficyClient.exe**.
6. Load the form in the client.

Generally, you will do this by running a workflow that has this form specified in a form activity. When the form is loaded into the Task List on this machine, you should see your breakpoints being hit in Visual Studio. Alternatively, you can select the form in the navigator, and open it in the **Global Viewer**. However, in this case no parameters will be passed into your form as they may be from a workflow.

7. Stop debugging.

Debugging this form should provide some information about what is going wrong. Before you make any changes, however, you must first stop debugging. Use the **Debug** menu in Visual Studio to stop debugging, and then make your code changes, save your changes, and recompile.

8. Reload the assembly.

Now that the assembly has changed, go to the Display Editor in the Client and upload the new version of the assembly. In addition, if you had added any additional files (including .pdb files), make sure you remove and re-add these to reflect the latest versions. If you had not specified .pdb files in this manner and had only manually copied them to the folder, you must copy them again to the `My Documents\Proficy Workflow\UI Assemblies` folder if you want to continue debugging.

9. Remove the symbol files.

When you are finished debugging, review Step 2 for instructions on removing the .pdb files when they are no longer needed.

Related information

[Form Editing and Debugging \(on page 1270\)](#)

Edit and debug content created in the Designer

This scenario deals with cases where you created a form using the Designer tab of the Display Editor in the Proficy Client.

Before you begin

Procedure

1. Create the form.

This is done using the Designer tab of the Display Editor.

2. Make the .PDB files available to the Client for debugging.

The .pdb file for the form in this scenario can be found at My Documents\Proficy Workflow\UI Projects\<formName>\bin\Debug\.

Making the .pdb files available for debugging can be done in one of 2 ways:

- Copy the .pdb files that correspond to the form assembly and dependencies to the My Documents\Proficy Workflow\UI Assemblies\ folder on the machine on which you want to debug the form.
 - When you are finished debugging the form and are ready to run in production again, delete these files from the My Documents\Proficy Workflow\UI Assemblies\ folder; otherwise, the symbol information will continue to be loaded each time the form is loaded.
 - Add the .pdb files as additional files in the **Display Editor** in the Workflow client. This automatically copies the file to the My Documents\Proficy Workflow\UI Assemblies\ folder for all machines when they try to load this form.
 - When you are finished debugging the form and are ready to run in production again, remove this as an additional file in the **Display Editor**. Be aware that this file will remain on any client machines that have already downloaded it, so you must also manually delete the file from the My Documents\Proficy Workflow\UI Assemblies\ folder. Otherwise, the symbol information will continue to be loaded each time the form is loaded.
3. Keep the source code local.

In the Designer, click the Project tab. Right-click the top node, which is labeled with the project name, and then select Delete local copy of this project when Editor is closed. This keeps the source code on this machine in order to enable debugging scenarios. Otherwise, the local copy of the source code is automatically deleted when you close the Display Editor.

The project is generated as a folder on the Client machine under <Logged in user> \My Documents \Proficy Workflow\UI Projects\AdjustTextSize. This folder can be downloaded onto the Proficy Client machine by opening the Client and loading the form into the Display Editor.

4. Open the project code using Visual Studio.

Open the code files you want to debug into and set your breakpoints by clicking in the extreme left margin. Putting a breakpoint in the constructor of your form is a good idea in order to test a breakpoint being hit.

5. Launch the Workflow client if it is not already running.
6. In the **Debug** menu of Visual Studio, select **Attach to Process**, and then select **ProficiencyClient.exe**.

Alternatively, if you want to debug the form from within the *Display Editor* while you are testing it, you can attach to FormTester.exe. This process appears only while a test is in progress.

7. Load the form in the client.

Generally, you will do this by running a workflow that has this form specified in a form activity. When the form is loaded into the Task List on this machine, you should see your breakpoints being hit in Visual Studio. Alternatively, you can select the form in the navigator, and open it in the **Global Viewer**. However, in this case no parameters will be passed into your form as they may be from a workflow.

8. Stop debugging.

Debugging this form should provide some information about what is going wrong. Before you make any changes, however, you must first stop debugging. Use the **Debug** menu in Visual Studio to stop debugging, and then make your code changes, save your changes, and recompile.

9. Reload the assembly and source.

Even though you have rebuilt the form locally, the changes have not been saved yet to the Proficity system. To do this, you must first open the form in the Display Editor and click the Designer tab. A prompt appears informing you that the source code saved for this form is different than the files on disk. Select No so that your local changes will be used and not overwritten. Your modified project is now loaded in the designer.

10. Save your changes to the server.

To save these source code files to the server, click the **Save** button on the title bar (in the upper right corner). If the **Save** button is not enabled, make a minor modification to the project to trigger an indication that it requires saving. Also, if you had added any additional files (including .pdb files), make sure you remove and re-add these to reflect the latest versions, and then save these changes, as well. If you had not specified .pdb files in this manner and had only manually copied them to the folder, you must copy them again to the `My Documents\Proficity Workflow\UI Assemblies` folder.

You can now test the new version of your form to verify that your changes have fixed the issue or go back to Step 4 to debug further.

11. Remove the symbol files.

When you are finished debugging, review Step 2 for instructions on removing the .pdb files when they are no longer needed.

Related information

[Form Editing and Debugging \(on page 1270\)](#)

WPF Forms Designer Tips for Editing and Debugging

This section provides hints and tips for improving and/or troubleshooting your WPF forms development.

Related information

[Form Editing and Debugging \(on page 1270\)](#)

If breakpoints are not hit when debugging

Breakpoints should be used and hit when debugging. If your breakpoints are not hit during a debugging session, follow the information below to troubleshoot the issue.

Does the <formname>.pdb file exist in My Documents\Proficy Workflow\UI Assemblies\ and does the timestamp on it match <formname>.dll? If not, choose one of the following methods to repair the problem.

- Update the <formname>.pdb file directly by copying it from My Document\Proficy Workflow\UI Projects\<formName\bin\Debug\<formName>.pdb
- In the Additional Files area of the Display Editor, remove and re-add the .pdb file, and then save the form.



Note:

If this file is modified on disk, the file saved to the server is not automatically modified. You must remove and re-add the file in the Display Editor.

The .pdb file is reloaded only if the .dll file has changed. If you modified the .pdb file but not the .dll file, restart the Workflow client and then re-attach to the Workflow client to debug.

Reloading forms into the Designer after editing in Visual Studio

In most cases, after a Designer-generated form has been edited in Visual Studio, the modifications can still be saved and then later edited in the Designer. However, there are exceptions where this may not work, such as in the following examples.

Related information

[Form Editing and Debugging \(on page 1270\)](#)

[Form Development in Visual Studio \(on page 1224\)](#)

Localizing a Form

If you want a form to always load in a specific language, you can set the language in code behind.

Use the following code snippets to set the form in a specific language. This example uses Spanish.

C#

```
C#: using LanguageSwitching;

LanguageSwitchingManager.Current.LanguageIdentifier = "Spanish";
```

VB

```
VB: Imports LanguageSwitching

LanguageSwitchingManager.Current.LanguageIdentifier = "Spanish"
```

Related reference

[Language Switching \(on page 957\)](#)

Context-aware displays

While forms are intended for use within a workflow, user displays are intended to live in the Workflow client, outside the context of a workflow.

User displays are created in much the same way as forms; however, there are limitations with user displays, as follows:

- Parameters on Dataltems must be configured at design time.
- The SourceAddress property (available only when the Advanced check box is selected) on a Dataltem can be used only for items with the same configuration. For example, on all Mixer classes that have a Speed property.

Validation on an output parameter

Use this information to help you configure a Masking TextBox control.

If the **IsRequired** property is set to true on a control's output parameter, the user will not be able to submit the form unless this output parameter has been set to any non-null value.

The user can submit a form only if the **IsValid** property of all output parameters evaluate to true. The **IsValid** property is typically configured with an expression.

The **InvalidText** property allows you to specify the error to display if a user attempts to submit the form when the **IsValid** value does not evaluate to true. If you do not specify an error in the **InvalidText** property, the expression configured for the **IsValid** property will be displayed.

Validation using the Masked TextBox

Use this information to help you configure a Masked TextBox control.

SectionTitle

The Masked TextBox control has all the usual TextBox properties with an additional **Mask** property. At configuration time you can enter the mask and specify the format of the mask. The following illustration provides an example.



The following illustration shows the text box at run-time. An underscore character () indicates that a letter or digit is expected.



Related reference

[Common Controls \(on page 758\)](#)

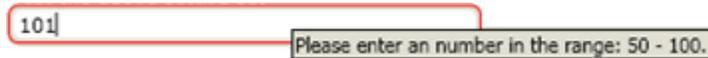
Validation using the Validating TextBox

Use this information to help you configure a Validating TextBox control.

The Validating TextBox control has all the usual TextBox properties with additional **IsValid** and **InvalidText** properties. The **IsValid** property is typically configured with an expression. The expression must evaluate to true for the value to be considered valid.

The **InvalidText** property is used to specify the tooltip text to display when the value entered in the text box is invalid.

When the data entry does not meet the validation specification, the text box outlined in red and a tooltip is displayed, indicating the error. The following illustration provides an example.



Related reference

[Common Controls \(on page 758\)](#)

Validation in code behind

Use this information to override the `ValidateForm` method, which prevents a form from being submitted by the user.

The following code snippet shows an example of how to achieve this.

```
public override bool ValidateForm(out string errorMessage)
{
    bool isValid = base.ValidateForm(out errorMessage);

    if (isValid)
    {
        // Do additional check here

        if (ValidatingTextBox1.Text == "55")
        {
            isValid = false;
            errorMessage = "Oh no! Not 55!";
        }
    }

    return isValid;
}
```

Adding Additional Assemblies to a Project

You can add a custom .NET assembly as a reference to the form project in the Project tab of the Designer.

If this assembly is not present on all client machines already, then it should be added on the Overview tab of the Designer to the **Additional Files** area. It is then downloaded to each client when the form is loaded.

On machines where the Designer will be used, the referenced assembly must also be copied into the Program directory of the Proficy install (as a peer to ProficyClient.exe). This is not required in cases where the Client is only running the form, but not modifying it.



Note:

Beginning with Workflow 2.5, .NET 4.5 is required to develop display or form assemblies. Custom display or form assemblies that target earlier versions of .NET will continue to function as before. However, saving changes made to such custom assemblies requires that they be upgraded, which is accomplished in different ways, based on the version of .NET:

- For an assembly that pre-dates .NET 4.0, you must upgrade the assembly when you open it. Failure to do so results in an error and the inability to save the edited assembly.
- For an assembly that targets .NET 4.0, the assembly is upgraded automatically when opened, and changes to such assemblies can be saved as before.

Related reference

[Designer: Project Panel \(on page 712\)](#)

Adding Controls to the Toolbox

If the newly referenced assembly contains any controls, they will automatically show up in the toolbox unless they have the [Browsable(false)] attribute specified.

You can specify which category heading any control should be shown under by using the attributes highlighted in the example below. You can also specify which heading a control's properties should appear under.

```
using System.ComponentModel;

[DisplayName("Masked Text Box")]
[DesignerCategory("Common Controls")]
[Description("The Masked TextBox.")]
public partial class MaskedTextBox : UserControl
{
    public static readonly DependencyProperty MaskProperty =
        DependencyProperty.Register("Mask", typeof(MaskString), typeof(MaskedTextBox),
            new FrameworkPropertyMetadata(new MaskString("AAA"),
```

```

new PropertyChangedCallback(OnMaskPropertyChanged));

    [EditorBrowsable(EditorBrowsableState.Always), Category("Validation"), Description("The mask which will determine
whether
the data entry is valid")]
    public MaskString Mask
    {
        get { return (MaskString)GetValue(MaskProperty) as MaskString; }
        set { SetValue(MaskProperty, value); }
    }

```

Using a Service Provider Method Exclusively in Code

Service provider methods, including those from custom service providers, are, in most cases, accessible and can be configured using the Server Method dialog box in the Designer. However, if a method is configured with the `Browsable` attribute set to `False`, then it will not show up in the Server Methods dialog box and can be accessed only through code.

The following procedure requires you to add additional assembly references to the form. Any additional references should be added through the Designer to ensure that they are added properly. If a reference is added in Visual Studio, be absolutely certain that you set the `CopyLocal` and `SpecificVersion` attributes to `false` or the form may not be useable with future product releases.

In forms or user displays generated in the Designer, you can access the `IServiceDirectory` service from the Host as follows:

```

IServiceDirectory serviceDirectory = this.Host.GetService(typeof(IServiceDirectory)) as
IServiceDirectory;

```

This requires an assembly reference to `Proficy.Platform.Core.ClientTools.Interfaces.IServiceDirectory.dll`, and the assembly containing the Service Provider interface you want to access. Additionally, you require a using statement for `Proficy.Platform.Core.ClientTools.ServiceDirectory.Interfaces`, as well as your Service Provider interface.

The `IServiceDirectory` service provides access to service providers through their interface names, as follows:

```

ICustomSP someCustomSP = serviceDirectory.CreateLogicalServiceProxy(typeof(ICustomSP).FullName)
as ICustomSP;

```

For example, an `IEquipment` Service Provider method can be executed with the following code:

```

IEquipment equipmentSP = serviceDirectory.CreateLogicalServiceProxy(EquipmentInterfaceNames.EQUIPMENT_IEQUIPMENT) as
    IEquipment;

Proficy.Platform.Forms.Editor.CustomControls.UniversalBrowser browser = UniversalBrowser1;

int result = equipmentSP.QueryPropertyCount(browser.SelectedItem as DirectoryResource, "%");

TextBox tb = TextBox1;

tb.Text = result.ToString();

```

To get this to work, you must add an assembly reference to Proficy.Platform.Services.EquipmentInterfaces.dll, and the following using statements:

- using Proficy.Platform.Services.Equipment.Interfaces;
- using Proficy.Platform.Core.ProficySystem.Types;

For a custom Service Provider, you require the appropriate assembly and namespace references.

Related reference

[Server and Client Methods Events \(on page 1147\)](#)

Sorting Items in a ComboBox, ListBox, or ListView

There is no property available on these Items Controls to enable alphabetical sorting of the contents; however, this can be done using code.

There are several ways to use code snippets to programmatically sort items in a control. The following is an example, written in C#.

In this example:

- `SearchMethod` is the name of the server method defined to execute the search method. An event handler was added to the ExecutedSuccessfully event in the Designer (that is, right-click the control, and then select **Add Event Handler**).
- `ComboBox1` is the name of the ComboBox.
- `DisplayName` is the column to sort.

```
using System;

using System.Collections;

using System.Collections.Generic;

using System.Data;

using System.Windows;

using System.Windows.Controls;

using Proficy.Platform.Core.DisplayFramework;

using Proficy.Platform.Core.DisplayFramework.Forms;

using System.Windows.Data;

using System.ComponentModel;

namespace SortedSearchExample

{

    public partial class MainForm : FormBase

    {

        public MainForm()

        {

            InitializeComponent();

            SearchMethod.ExecutedSuccessfully += new System.EventHandler(this.SearchMethod_ExecutedSuccessfully);

        }

        private void SearchMethod_ExecutedSuccessfully(object sender, System.EventArgs e) {

            Sort("DisplayName", ListSortDirection.Ascending);

        }

        private void Sort(string sortBy, ListSortDirection direction)

        {

            ICollectionView dataView = CollectionViewSource.GetDefaultView(ComboBox1.ItemsSource);

            if (dataView != null)

            {

                dataView.SortDescriptions.Clear();

                SortDescription sd = new SortDescription(sortBy, direction);

                dataView.SortDescriptions.Add(sd);

                dataView.Refresh();

            }

        }

    }

}
```

}

Chapter 10. Building and Configuring Workflows

Workflow Design Basics

Workflow Design Basics

Workflows are configured collections of subprocesses, activities, reference data, and functional resources that are used to complete the defined process from start to finish.

Under the Workflow model, using the Workflow Editor, you configure all aspects of a workflow's processes and specifications that are used during production.

Workflow logic may be internal and invisible to operators; however, most of the logic requires interaction through a user interface. Workflows that are visible to operators are called tasks, and when they are running, they can be viewed in the Task List or Task Management display, depending upon the permissions available to a user.

A workflow may also be composed of subprocesses. These subprocesses may only be internal and invisible to operators; however, they must be valid (that is, contain no errors) and enabled (active) for a workflow to run. But, like workflow, they may require user interaction. If they are visible to operators, they are called task steps. Task steps can include their own documentation and forms, which appears to users in the Task List display.

Tasks and task steps can be routed to different workstations based on equipment or user requirements. For example, a task might contain an Insert Flange task step. When executed, this task step is routed to the console of the flange insertion worker. This task step presents him with a diagram of the flange install and a form to submit when he is finished. Sequences of task steps can be used to guide users through complex operations. Operators can input data into forms, that can then be used by the rest of the workflow or stored for reporting purposes.

Bind a resource property to a workflow or subprocess

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. To configure a workflow:

- a. Click the arrow next the property you want to bind.
 - b. Click **Bind**.
The **Select Variable** dialog box appears.
5. To configure a subprocess:
- a. From the subprocess configuration panel, click **Override Workflow Task Assignments**.
 - b. Click the arrow next the property you want to bind.
 - c. Click **Bind**.
The **Select Variable** dialog box appears.
6. Locate and select the resource you want to use, and then click **OK**.

**Note:**

Only one resource can be selected.

7. Click **Save**.

Add an expression to a workflow or subprocess

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the arrow button next to the property you want to add an expression to, and then select **Expression**.
The **Expression Editor** appears.
5. In the **Enter Expression** field, build the expression by selecting expression functions, expression variables, and/or expression operators, and then enter user-defined values.
6. To check the expression and condition for configuration errors, click **Check Syntax**.
7. To evaluate the expression, click **Test**.
8. Click **OK**, and then click **Save**.

Remove a bind, expression, or property value from a workflow or subprocess

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource type list, select a resource.
3. In the Displays panel, click **Workflow Management > Workflow Editor**.
4. On the workflow configuration panel, click the arrow next to the property you want to work with:
 - If a resource is selected, click **Reset to Default**.
 - If a resource is in a bind or expression, click **Literal**.
5. On the subprocess configuration panel, click the arrow next to the property you want to work with:
 - If a resource is selected, click **Reset to Default**.
 - If a resource is in a bind or expression, click **Literal**.
6. Click **Save**.

Workflow Binds

A binding allows you to specify the source from which the actual value will be obtained. When the value for the source changes, the bound property value will automatically update, as well.



Important:

Bind values must match the displayed data type.

Forms and Binds

When you perform binds on form control properties, the above information is also true, with the addition of the following information.

For some properties, you can also create a two-way binding, whereby if the bound property value changes, the source also changes. This type of scenario is property dependent; that is, it does not apply to all properties on all form controls.

Bindings that involve input parameters or output parameters are explicitly created as one-way bindings. For example, if you bind from an output parameter's **Value** property to the **Text** property of a text box, then if the text in the text box changes, the output parameter updates. However, if the output parameter's **Value** property changes, the text box's Text property does not update.

Workflow Expressions

A workflow expression is an algebraic expression that defines a calculation that the system performs at run time. Workflow expression syntax allows you to manipulate strings, test for relationships and conditions, and perform arithmetic calculations.

You can build simple or complex expressions. They are user-defined and consist of one or more of the following components:

- Functions
- Operators
- Variables
- Literals

Workflow Expression Editor

The following table displays the attributes that are available in the workflow expression editor:



Tip:

For general information on building expressions, see *Expression Editor*.

Attribute	Value	Description
Expression Functions tab	System-defined	Displays the functions available to build your expression. Double-click a function to insert it into your expression.
Workflow Variables tab	User-defined	Displays the variable browser where the variable properties are grouped by a containing activity and a property category. You can use variables to build your expression. The variables available are dependent on where you are located within the workflow configuration process (such as inside an activity or sub-process), and which resources you have already added to the workflow. These resources include:

Attribute	Value	Description
		<ul style="list-style-type: none"> • workflow-level variables • subprocess-level variables • activity-level variables • user activity-level variables • fault handler activity variables <p>Select a variable property to display the following information related to it, and double-click it to insert it into your expression:</p> <ul style="list-style-type: none"> • data type • bind path • description <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  <p>Note: The default value for date/time is always associated with the <i>current time zone</i>, unless a time zone is specified using a different function in the expression. Any actions performed in the same expression using this resultant value uses the current or specified time zone for evaluation.</p> </div>
Completion Codes tab	User-defined	Displays the completion codes available to build your expression.

Attribute	Value	Description
		Double-click a completion code to insert it into your expression.

**Note:**

The property you are creating an expression for and its data type are displayed in the Expression Editor in the upper left and upper right, respectively.

Workflow Literals

A literal value allows you to enter a static value for the resource you are configuring.

**Important:**

Literal values must match the displayed data type.

Workflow Properties

You can use properties to further define the resource you are configuring.

Properties existing in the workflow or process scopes that are defined on the workflow are available when authoring a global subprocess. These new properties are different from other workflow variables like local variables or parameters because they are available on every (process) workflow definition.

Properties

The following table describes the available options.

Property	Read	Write	Description
Absolute Expiry Time	All	None	In a preprocess or a pre-subprocess, specifies an absolute expiry time (DateTime value) if the Expiry Type is set to Absolute.
Relative Expiry Time	All	None	In a preprocess or a pre-subprocess, specifies a relative expiry time (TimeSpan value) if the

Property	Read	Write	Description
			Expiry Type is set to Relative.
Expiry Type	All	None	Specifies the expiration as Absolute, Relative, or None during run time.
Auto Select	All	None	Flags a task to be executed prior to non-flagged tasks in a workflow during run time.
Id	All	None	Read Only Unique key that identifies the workflow definition in the data model.
Instance Id	All	None	Read Only Unique key that identifies the workflow instance in the data model and in the engine.
Name	All	None	Read Only Display name of the workflow definition resource.
Schedule Id	All	None	Read Only Unique key that identifies the workflow schedule in the data model. This property has a value only if the workflow instance starts by a schedule.
Schedule Name	All	None	Read Only

Property	Read	Write	Description
			Display name of the workflow schedule resource. This property has a value only if the workflow instance starts by a schedule.
Started By Schedule	All	None	<p>Read Only</p> <p>Display name of the workflow schedule resource. This property has a value only if the workflow instance starts by a schedule.</p>
Subprocesses	All	None	<p>Read Only</p> <p>Collection of all of the subprocesses contained in the workflow. This property allows you to access the properties of any subprocess contained in the workflow using the workflow variable viewer.</p>
Task Instance Id	All	None	<p>Read Only</p> <p>Unique key that identifies a task (instance). Task instance IDs are assigned each time a process loop begins; that is, just before the preprocess executes, and clears when the task completes or is can-</p>

Property	Read	Write	Description
			called after the post-process completes.
Task Instance Name	All	All	String name of the task instance.
Task Instance Priority	All	All	Priority of the task instance in number format.
Task Instance Expired	All	All	Time that the task instance can no longer be performed.

Limits for Workflows

There are restrictions to the elements that comprise a workflow. These limits ensure a quality functioning workflow.

You can create a maximum of 1000 workflows; however, the size of each workflow can affect the performance of your system.

Activities, local variables, and expressions (used for conditions, bindings, and so on) all contribute to the size of a workflow. The type of activity also contributes; for example, a call method uses more memory to initialize than a simple write activity does. Each branch of an If/Else or parallel activity is considered an activity. For example, an If/Else activity with two branches is considered three activities before you add anything to either branch.

Keep the following guidelines in mind when you are building workflows.

- Keep the total activity count to fewer than 2000 (across all subprocesses)
- Keep the total number of variables, parameters, external events, and data items to fewer than 500 (across all subprocesses)



Note:

Global subprocesses and user activities each count as single activities regardless of the number of child items they contain. These activities can be used to help reduce the size of your workflows.

**Important:**

Data item limits include child activities.

Limit	Restriction	Result
Data items and External events (including contained activities)	100	Validation error.

Workflow Editor User Interface

The **Workflow Editor** tab displays the graphic interface for creating and editing a workflow, as well as navigating between panels during configuration.

Cut, copy, and paste in the Workflow Editor workspace

Before you begin

**Note:**

This procedure applies to subprocesses and subprocess activities.

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. In the workspace, select the activity you want to cut or copy.
6. On the toolbar, click either **Cut** or **Copy**, depending on what action you want to perform.
7. Click in the workspace, and then click **Paste**.

The object is pasted onto the last drop point.

Delete in the Workflow Editor workspace

Before you begin

**Note:**

This procedure applies to subprocesses and subprocess activities.



A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. In the workspace, select the activity you want to cut or copy.
6. On the toolbar, click either **Cut** or **Copy**, depending on what action you want to perform.
7. Click in the workspace, and then click **Paste**.
The object is pasted onto the last drop point.
8. In the workspace, select the subprocess you want to delete or open the subprocess containing the activity you want to delete.
9. On the toolbar, click **Delete**.

Undo and redo in the Workflow Editor workspace

Before you begin



Note:

If you navigate from the editor and then return, you cannot undo or redo previous tasks in the workflow.

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. In the workspace, select the activity you want to undo or redo.
6. Click **Undo** or **Redo** on the toolbar to undo a change.

Click...	To...
Undo	undo a change you made to the workflow or subprocess.
Redo	re-apply a change that was just undone.

Print a workflow

Before you begin



Note:

If you navigate from the editor and then return, you cannot undo or redo previous tasks in the workflow.

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. On the toolbar, click **Print**.
The Print dialog box appears.
6. Select the appropriate print settings.
7. Click **OK**.

Perform Print Preview

Before you begin



Note:

If you navigate from the editor and then return, you cannot undo or redo previous tasks in the workflow.

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. On the toolbar, click **Print Preview**.
The workspace view changes to print layout.

Perform Page Setup

Before you begin

In Page Setup, you can format the following:

- paper size and source
- page orientation and view
- margins
- page scales
- header and footer

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. On the toolbar, click **Page Setup**.
The Page Setup dialog box appears.
6. Select the appropriate page and header/footer settings.
7. Optionally, click **Printer** to set printer settings and to print.
8. Click **OK**.

Pan the view in the Workflow Editor

Before you begin

If the current workflow or subprocess exceeds the boundaries of the **Workflow Editor**, scroll bars appear.

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. On the vertical scroll bar, click **Pan**.
6. Do one of the following:

Do this...	To...
Zoom In	increase the size of the workspace
Zoom Out	decrease the size of the workspace
Navigation	manually move the workflow within the workspace
Default	return the workspace to its original size

Zoom the view in the Workflow Editor

Before you begin



Note:

This procedure applies to subprocesses and subprocess activities.

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. On the vertical scroll bar, click **Zoom Level**.
6. From the menu, click the percent of the workspace that you want to see. The default is 100%.

Results

**Tip:**

Click **Show All** to automatically fit the workflow to your current view.

Workflow Foundations

Workflow Foundations

This section provides you with general information about workflows, from information you need to develop them to running them.

Tasks

Workflow tasks are accessible when a workflow has been put into a run time state. When a workflow is running, the task information is available to the computers and operators assigned to that task or task step, as well as to administrators.

Tasks can be broken down into multiple task steps by defining subprocesses that contain specific information on how a task should be completed. If a workflow is set as a continuous workflow, then each running loop finishes the post process and starts a completely new running workflow within the Task List.

When a workflow is created, enabled, and saved, it is available to be executed. Part of creating a workflow includes determining operator tasks, such as deciding task priorities, writing work instructions, linking required documents, assigning personnel and equipment, configuring conditional events, and setting task expiration. These tasks are defined in the *Workflow Editor*.

Task step information is displayed in the Task List in *General Operating Environment* or in *Task Management for Administrators* while a task is being processed. Users, such as operators or administrators, can see all tasks that need to be performed for a running workflow. In the Task List, an operator can filter the tasks or task steps to include only his own, or he can view all tasks and task steps assigned to him. Higher level users, such as administrators, can view this same information in Task Management.

**Tip:**

The Workflow Task List can be integrated into other GE HMI applications. For more information, see Task Controls in GE HMIs.

Workflow Execution Engine Overview

This application consists of a workflow execution engine, which is the core component of the process management system. The engine is responsible for interpreting workflows, controlling multiple workflow instances, and sequencing subprocesses and activities.

These execution control functions invoke the appropriate human, equipment, and application resources necessary to complete a process. They act as the link between the process as modeled within the workflow and the process as it is seen in the real world.

The workflow execution engine is also responsible for starting and ending a workflow instance, controlling activity scheduling within an operational process, interacting with applications, process equipment, and human resources. This can be distributed across a number of computing resources to cope with the volume of processes that operate within a facility.

The execution engine records internal *tracking data in workflows* that includes the state information associated with a running workflow instance. The execution engine persists the workflow instance state when a subprocess completes. The persisted state can be used to recover and restart a workflow instance from failure conditions, or if it was stopped. If a workflow is stopped normally, the postsubprocess, postprocess, and unload steps run before the instance stops (they will not run if the server crashes). The workflow instance can continue from the start of the current subprocess by using the restart command. When this happens, the instance restarts by running the load, preprocess, and presubprocess steps. Then it restarts at the beginning of the subprocess that was running before the workflow stopped.

You can interact with the execution engine to monitor, troubleshoot, and debug a running workflow. This is a critical feature that allows you to identify and diagnose any issues with a workflow or any of its activities to work towards completion of the instance.

Persistence in Workflows

The execution engine records internal *tracking data in workflows* that includes the state information associated with a running workflow instance. The execution engine persists the workflow instance state when a subprocess completes. The persisted state can be used to recover and restart a workflow instance from failure conditions, or if it was stopped. If a workflow is stopped normally, the postsubprocess, postprocess, and unload steps run before the instance stops (they will not run if the server crashes). The workflow instance can continue from the start of the current subprocess by using the restart command. When this happens, the instance restarts by running the load, preprocess, and presubprocess steps. Then it restarts at the beginning of the subprocess that was running before the workflow stopped.

Building and Configuring Workflows

Workflows consist of programs that can carry out complex activities, respond to events and data changes, and write values back out to the data models or through external connectors.

You can define a process or workflow, which is normally comprised of a number of steps with a logical execution path. Each step can involve manual interaction with a user or can be automated with your equipment. This software enables your process to follow and record the outcome of the execution.

Workflows are configured collections of subprocesses, activities, reference data, and functional resources that are used to complete the defined process from start to finish.

Workflow Foundations provides you with general information about workflows, from information you need to develop them to running them.

The Task Management display shows task and task step information related to the corresponding Equipment, Personnel, or Proficy System Computers as selected in the navigator.

The workflow configuration panel in the Workflow Editor allows you to configure task details, properties, parameters, variables, events, namespaces, and data sources for workflows.

A subprocess is part of a user-defined process in a workflow. For a workflow to run, all subprocesses must be valid (contain no errors) and enabled (active).

Global subprocesses allow you to develop and store standard and frequently used subprocesses. These subprocesses can be reused in various workflows. They are preconfigured and then are available in the Global Subprocesses list where you are creating your workflow.

When a global subprocess is inserted into a workflow, you can change it to a local subprocess. Then, you can configure the subprocess to meet the specific needs of the currently selected workflow.

Activities comprise the main building blocks or steps that perform the actions in a subprocess or workflow.

Schedules are used to run a workflow when an event occurs and provide more configuration options for specific workflows; however, they are created and executed separately from workflows.

Faults are errors or failures that occur within a workflow while it is running.

You can test and review workflows before they become part of your production environment by performing basic tests and verifying instance control.

The debugging function allows you to locate and isolate an error in your workflow in order to fix it before the workflow is used in a production environment.

The Tracking Configuration tab allows you to track subprocess or workflow level data and local variables while a workflow is running.

The Web Services Service Provider extends Workflow capabilities by allowing you to use SOAP (Simple Object Access Protocol) web services that provide WSDL (Web Services Description Language) metadata.

Workflow publishes a web service, IWorkflowWebService, that allows schedule operations. You can browse all configured workflow schedules and start one or more of them, as well as get information on running workflow instances.

Categories allow you to organize the resources associated with your workflows.

Completion codes indicate how a subprocess finishes, and may change depending on the outcome of the subprocess execution.

Resource actuals show the as-performed actual values for a resource during a production cycle. A resource actual is associated with a resource specification, making it easy to compare actual and specified values for properties.

The Workflow Task List can be integrated into other GE HMI applications.

Service interfaces allow you to access one or more call methods that you can use when developing your workflow.

eSOP (Electronic Standard Operating Procedures) offers a simplified solution for managing and developing workflows. It increases the efficiency of your development process by utilizing differing skill levels to develop the final workflow product.

You can use the REST APIs to interact with the Workflow application over the Internet.

Process Execution Model Overview

The process execution model defines the structure of a workflow, and has several process steps that can be programmed and triggered during a workflow instance.

All workflows are process-oriented. A workflow is a representation of what happens as the result of executing the process. It is made up of subprocesses, which are made up of activities. A subprocess is a grouping of activities that accomplish a specific outcome. An activity is a single logical action within a subprocess and is the smallest unit of work that can be performed within a process.

A scheduled task contains all necessary information about the process to enable it to be executed by the workflow execution engine. A scheduled task specifies when a process begins (that is, when it enters a run-time state, or becomes a workflow instance), the event that triggers it, who the instance is assigned to, the specific location it must be performed, or what inputs/outputs are required.

When the workflow definition begins running, the load step starts and any activities within this step run. After the load step runs, the preprocess step is triggered, which begins the process event cycle.

When the preprocess step finishes, the body of the process begins. The body consists of the presubprocess and postsubprocess steps, as well as a subprocess and flow control activities such as *If/Else* and *Parallel* activities. A process can also consist of one or many subprocesses and each subprocess has zero or more activities grouped together within it. The cycle of presubprocess, subprocess, and postsubprocess run until all the subprocesses within the process are complete. The postprocess step begins after the last subprocess ends. When the postprocess step ends, the unload step begins. When this step finishes running, the workflow instance ends.

Workflow Process Steps

Process steps are part of every process-oriented workflow.

The standardization of these process steps is important because it gives you the flexibility to define a workflow and each step in a manner that is most effective and relevant to a specific process and the needs of your business. However, for a workflow as a whole to be functional, you must add activities into the process steps. See *Activities Used to Create Workflows* for information on building your workflow definition.

The process steps described below run during a workflow instance at least once; however, the process flow is modified when:

- an unhandled fault occurs;
- the workflow is configured to run continuously; or
- the workflow instance stops manually.

When an unhandled fault occurs, the current subprocess ends at the fault, but the workflow instance ends in the normal way. That is, the postsubprocess, postprocess, and unload steps run before the instance completes.

When a workflow runs in a continuous loop, the load step runs once during its first execution cycle. For subsequent cycles, the workflow repeats the process cycle starting with the preprocess step and continues through the postprocess step. The load step does not run again. The continuous loop cycle

continues in this way until the process is shutdown or an unhandled fault occurs. At this point, the unload step runs and the workflow instance ends.

A workflow instance can also be stopped while it is running. The workflow is halted in a controlled manner, but it is still live, which means that it is still available to run by restarting it.

Load

The load step is the first step in a workflow cycle. It is used to perform a one-time only setup for the workflow instance.

The load process is a good starting point to add initial data common setup procedures inherent to your process. It is also a good starting point from which to facilitate the completion of other subprocesses located within the workflow instance. By using the load step in this manner, you can ensure that the necessary steps for the process are enabled and executed.

If the workflow is set on a continuous loop, this step only runs once.

Preprocess

The preprocess step starts after the load step ends. This step performs any initialization work required for the process.

If the workflow is set on a continuous loop, this step becomes the first step reinitiated after the postprocess step ends. The continuous loop then runs between the preprocess and postprocess steps. The only time this loop breaks is if an end command is used or an unhandled fault occurs. The workflow then continues through the rest of the steps to the unload step.

Presubprocess

This step performs any initialization work required for the subprocess.

The presubprocess step starts after the preprocess step and at the beginning of every subprocess. It allows a variety of input and output scenarios to be completed before each subprocess starts.

The following table describes the available options.

Property	Read	Write	Description
CurrentSubprocess	N/A	N/A	Read Only Collection of all of the subprocesses contained

Property	Read	Write	Description
			in the workflow. This property allows you to access the properties of any subprocess contained in the workflow using the workflow variable viewer.

Task Step (Subprocess) Configuration

A subprocess is part of a user-defined process in a workflow.

For a workflow to run, all subprocesses must be valid (contain no errors) and enabled (active).

A workflow must be constructed of one or more subprocess. Each subprocess is a self-contained entity where the majority of the logic is added to run a workflow. Subprocesses group a set of related activities in a workflow and appear as a single task step. A subprocess can exist only between the presubprocess and postsubprocess.

Subprocesses allow you to create a series of steps to accomplish a task, assignment, event-driven, or activated process. The contents of a subprocess consist of activities that can be placed in any order depending on the requirement. You can think of the contents of a subprocess as a workflow without the defined structure and in a simple linear path.

Within a subprocess activity, you can select a manual start option for a workflow. This option allows you to configure a subprocess to be either an automated aspect of a workflow, or to require operator interaction within the workflow process. When the manual start check box is cleared, the subprocess runs automatically when the workflow is activated. When this check box is selected, the workflow stops when a manual start subprocess is reached within the workflow. At this point, an operator is required to intervene in the process and complete an action before the workflow can continue.

Within a workflow definition, you can select a continuous option for the workflow instance. However, in a continuous loop, you cannot loop a single subprocess. During the continuous loop, every step from the preprocess to the postprocess is completed in the continuous loop cycle. You can, for example, put in conditional activities to facilitate a small loop inside of the subprocess in case a step is not completed or an error found.

Inputs and outputs of a subprocess can be linked from and to other subprocesses or execution cycle steps. For example, you can create a dependency on the output of subprocess1 for subprocess2, or the

input of subprocess1 is dependent on the output of the preprocess or load steps. Therefore, you can ensure that at each step of the execution cycle, there are dependencies that must be met before the cycle can continue onto the next step.

Subprocesses also have a built-in completion code variable that is always tracked. While a subprocess is running, this variable can be set to any available completion code for a project. A default value must be supplied for this completion code.



Note:

You can configure up to 1000 subprocesses.

Postsubprocess

The postsubprocess step starts after the subprocess, and at the end of every subprocess listed.

This step performs any uninitialization and recording work required for the subprocess. It allows a variety of input and output scenarios to be completed after each subprocess ends. At the end of this step, tracking data for the subprocess is recorded, including the completion code and any tracked subprocess variables.

Properties

The following table describes the available options.

Property	Read	Write	Description
CurrentSubprocess	N/A	N/A	Read Only Collection of all of the subprocesses contained in the workflow. This property allows you to access the properties of any subprocess contained in the workflow using the workflow variable viewer.

Postprocess

The postprocess step completes a workflow before the unload step starts.

The postprocess activities are identical to any subprocess, but the setup of these activities is different because it is the end of the workflow cycle. At the end of this step, tracking data for the process is recorded, including any tracked process variables.

If the workflow is set on a continuous loop, this step is the last step. The workflow runs through the postprocess and then starts again at the preprocess step. The only time this loop breaks is if an end command is used or an unhandled fault occurs. The workflow then continues past the postprocess step to the unload step.

Unload

The unload step is the last step in a workflow cycle.

When the unload step is triggered and its activities are finished running, the instance ends.

Resource View

The Resource View provides you with more information at a glance of certain properties associated with workflow and schedule resources.

These properties are in a table that you can sort by clicking on a column heading.



Tip:

To view the properties associated with workflow resources, select View Workflows. To view the properties associated with schedule resources, select View Schedules.

Attribute	Resource	Description
Name	Workflows and Workflow Schedules	The name given to a resource when it was created. <ul style="list-style-type: none"> To view and edit this resource, select the name. The resource opens in the Workflow Editor.
Description	Workflows and Workflow Schedules	The information provided in the description field of your resource.
Enabled	Workflows and Workflow Schedules	Corresponds to the activated state of workflows or workflow schedules.

Attribute	Resource	Description
Count	Workflows	<p>Indicates the number of running instances for all versions of this resource.</p> <ul style="list-style-type: none"> To view this resource, select the number. The resource opens in the Running Workflows.
Version	Workflows and Workflow Schedules	<p>Indicates the current user-defined version of the resource being used.</p> <div data-bbox="1024 800 1417 1020" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: This section indicates the latest version used, not all versions available.</p> </div>
Last Modified	Workflows and Workflow Schedules	Indicates the date and time the resource was last modified.
Category	Workflows and Workflow Schedules	Shows the name of any categories that are assigned to the resource.

Task Management for Administrators

Task Management for Administrators

The Task Management display shows task and task step information related to the corresponding Equipment, Personnel, or Proficy System Computers as selected in the navigator.

This information is generally used by administrators for high-level navigation. A user with the appropriate security permissions can view all tasks and task steps routed to any selected location, personnel, or resource. For example, all instances of a specific computer being used to perform certain tasks. As a result, administrators can oversee all running tasks.

The Task Management display appears identical to the Task List; however, there are actions you can perform and displays that are visible only in the Task List. For example, the ability to acquire tasks and provide input through the Task Inputs tab is only available in the Task List.

Information that is displayed only in Task Management include tasks that have missing resource information, such as equipment or personnel assignments.

Workflow Instance Management

You can view the status of your workflow, its resources and the number of these resources currently running. This information is displayed in the **Overview** section when you are configuring your workflow.

Run-time Status

A workflow whose status is set to `Enabled` can be used in a live production environment. All resources, such as schedules, associated with that workflow must also be enabled. A workflow whose status is set to `Disabled` cannot be used in a run-time state; however, it can be edited and run in test mode.

**Note:**

If a workflow is a duplicate of another enabled workflow, it must be enabled again after it is duplicated.

Instance Behavior

You can define the allowed behavior of a workflow. Select from the following options:

- Allow multiple instances of a workflow to run at the same time.
- Disable persistence for instances of a workflow.

**Tip:**

Persistence is disabled by default, which provides better performance of your workflow. To support recovery in the event of server problems, or to stop and restart your workflow, enable persistence by removing the selection from this check box.

- Automatically restart stopped instances of a workflow when the server start.

Related information

[Workflow Execution Engine Overview \(on page 1299\)](#)

[Persistence in Workflows \(on page 1299\)](#)

Configure workflow or schedule instance behavior

Procedure

1. Select the applicable resource from one of the following locations:

For...	Do this...
Workflows	In the navigator, click Workflow > Workflows .
Schedules	In the navigator, click Workflow > Schedules .

2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. On the **Overview** tab, in the **Workflow Instance Behavior** section, select the appropriate check box for your workflow.
 - To run more than one of the same workflow at the same time, select Allow multiple instances of this workflow to be running at once.
 - For better workflow performance, select Disable persistence for instances of this workflow.
 - To restart stopped instances, select Automatically restart stopped instances of this definition on server start.
5. Click **Save**.

Enable and disable a workflow

You can edit an enabled workflow for it to be available the next time a workflow runs.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. On the **Overview** tab, in the **Status** section, perform one of the following tasks:

Click...	To...
Enable	activate the workflow, and to use it with a workflow schedule.
Disable	deactivate the workflow.

Workflow Instances View

Workflow Instances is an interactive area that allows you to perform various actions, such as viewing, monitoring, controlling, and debugging workflow instances.

In the workflow display, you can choose a specific workflow instance to view, or you can view all instances of the selected workflow. If you select **Show all instances**, information for all current workflow instances displays. If you select **Show instances of selected object**, information for only the selected workflow instance displays. This display also provides information on the current runtime and debug state of the workflow instance.

You can use the headings of this display to sort workflow instances. For example, if you click on the State heading, all workflow instances are sorted alphabetically by state.

Instance Details

This section provides general information about a selected workflow instance, such as the Instance ID, the name of the workflow definition and the schedule it is attached to, the date and time the instance was started, as well as how it was started. The values of input parameters passed to this workflow are also listed in this section.

Instance Control

This section allows you to interact with a selected workflow instance. You can control it through the following commands:

- **Stop**

Use this button to halt the workflow by cancelling the current subprocess and executing the post-process steps.

- **Restart**

Use this button to reset the workflow instance at the start of the current subprocess and continue running it. You can use this button to restart a workflow if it has not been configured to restart automatically.

- **Terminate**

Use this button to end the workflow instance at the current execution point. The post-process steps do not run and the instance cannot be restarted.

Debugging Control

This section allows you to start a debugging session for a workflow instance. For more information on these commands, see Debugging a workflow instance.

Stop and restart a workflow instance

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
3. From the Workspace, select the workflow instance you want to work with.
4. To stop the workflow, click **Stop**.
5. To restart the workflow, click **Restart**.

End a workflow instance

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
3. From the Workspace, select the workflow instance you want to end, and then click **Terminate**.

Compile Errors

Compile errors occur when there is a problem with a workflow and it cannot execute.

Workflow Errors

When a workflow resource is invalid, such as a subprocess or user activity, you receive an error message that you can view in the Overview section of the Workflow Editor, or in the Feedback Zone. The type of errors you see can be either configuration errors, such as a broken binding, or external errors, such as a missing resource being referenced in the workflow (for example, a data item). You can use the compile feature to resolve an issue with external resources only. For example, if you restore a missing piece of equipment, you can use the Compile button to make the workflow valid.

The following table describes the available options.

Attribute	Value	Description
View Errors	N/A	Click to view errors related to the current workflow.

Attribute	Value	Description
Compile	N/A	Click to recompile a workflow after you resolve the error.

Compile workflows to fix errors

To view errors associated with a specific workflow, select that workflow.

Before you begin



Note:

You can use the compile feature to resolve an issue with external resources only.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. On the **Overview** tab, you can view the errors, if any, associated with the selected workflow. To see these errors, click the **View Errors** button.
5. After you fix any external errors associated with the workflow (such as data items, events, or missing equipment), click the **Compile** button.
6. Click **Save**.

Version Control

You can update workflow resources (workflows, subprocesses, user activities, and schedules) while a workflow is running. This information is displayed in the **Overview** section when you are configuring your workflow.

You can modify enabled workflow resources without stopping all of the running workflows that are associated with those resources. As a result, you can make changes that affect an existing workflow both in a development and production environment. All existing running workflows will continue to run with the older version of the workflow until that workflow has terminated. Any newly started workflows will use the new revision.

Related reference

[Workflow Instance Management \(on page 1308\)](#)

Related information

[Add manual version control to a workflow, subprocess, user activity, or schedule \(on page 1313\)](#)

Add manual version control to a workflow, subprocess, user activity, or schedule

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource type list, select a resource.
3. On the **Overview** tab, in the **Version** field, enter a value.
4. Click **Save**.

Related reference

[Workflow Instance Management \(on page 1308\)](#)

[Version Control \(on page 1312\)](#)

Task Configuration

Task Configuration

The workflow configuration panel in the **Workflow Editor** allows you to configure task details, properties, parameters, variables, events, namespaces, and data sources for workflows.

Under the **Task Details** tab in the **Workflow Configuration Panel**, you can input or attach more information related to a workflow that helps a user perform tasks in the Task List or Task Management.



Important:

To validate the workflow, a location and/or personnel task assignment must be selected.

Developing a web form for a workflow

This tutorial shows how you can design, develop, and use a web-based form, then add it to a simple workflow for use by an operator working in the Task List.

Before you begin

The following procedures indicate the basic steps involved in creating a simple workflow with a web form.

- Log in and launch Workflow client
- Create and configure a web form with appropriate labels and operator controls

**Tip:**

Text fields can be used to display read-only information, such as headings or titles on your form. Text boxes can be used for processing operator input.

- Create and configure a workflow
- Add a form to a workflow
- Add activities to a workflow to demonstrate binding user input to a form
- Add instructions for operators
- Create a schedule to automatically start a workflow
- Test the workflow in the Task List

About this task**Note:**

For a video tutorial of this topic, use a browser to open <http://www.youtube.com>, and then search for GE Proficiency Workflow - Developing a Web Form for a Workflow.

Procedure

1. Log in to Workflow.
 - a. Launch Workflow client using either the **Start** menu or desktop icon.
 - b. On the main Workflow screen, click . The **Workflow Login** dialog box appears.
 - c. **Optional:** In the **Authentication Type** section, select the type appropriate to your user profile.
 - d. In the **User Name** field, enter your user name.
 - e. In the **Password** field, enter your password.
 - f. Click **OK**.
2. Create a web form resource.
 - a. From the navigator, click **Global Displays > Forms**.
 - b. Click **Add Form**.
 - c. In the **Name** field, enter a name, such as `My First Form`.
 - d. In the **Description** field, enter clarifying information about the web form resource, such as `This is my first web form`.

- e. Click **OK**, or press `ENTER`.
 - f. In the **Displays** panel, click **Display Editor**.
 - g. On the **Overview** tab, select **Use the Designer tab to create a simple web form**.
 - h. Click **Save**.
3. Design a form by adding a read-only text field to the web form.
 - a. Click the **Designer** tab.
 - b. In the **Form Fields** section, click the **Text** form control to add the field to your form.

**Tip:**

You can add multiple text fields, as well as other form controls. Each text control is labelled and numbered in the **Name** field in sequence; for example, *Text1*, *Text2*, and so on.

To delete a field, select it, and then click **Delete**.

The **Layout** area displays the basic configuration for your form.

- c. In the **Properties** section, you can configure the form fields.
 - i. In the **Name** field, enter a unique identifier by modifying the existing text. For example, enter `Text Control Name`. For example, enter `Text Control Name`.
 - ii. In the **Text** field, enter content that is displayed to the operator and provides more information about the form. For example, enter `Form Title`.
 - iii. In the **Size** field, from the drop-down list, select a text size.
 - iv. In the **Alignment** field, from the drop-down list, select an alignment value.
4. Design a form by creating a text box to the web form.
 - a. In the **Form Fields** section, click the **Text Box** form control to add the field to your form. The **Layout** area displays the basic configuration for your form.
 - b. In the **Properties** section, you can configure the form fields.
 - i. In the **Name** field, enter a unique identifier by modifying the existing text. For example, enter `Text Box Control Name`.
 - ii. In the **Label** field, enter descriptive text on the form control that is displayed to the operator. For example, enter `Text Box with Status Message`.
 - iii. From the **Type** drop-down list, ensure **Text** is selected.

No further configuration is required.

- c. Click **Save**.
5. To view your form or display, in the **Displays** panel, click **Global Viewer**.
 6. Create a workflow resource.
 - a. In the navigator, click **Workflow > Workflows**.
 - b. Click **Add Workflow**.
 - c. In the **Name** field, enter a name, such as `My First Workflow`.
 - d. In the **Description** field, enter clarifying information about the workflow resource, such as `This is my first workflow`.
 - e. Click **OK**, or press `ENTER`.
 7. Configure the workflow resource.
 - a. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
 - b. Click the **Workflow Editor** tab.
 - c. To assign a task in the workflow, on the **General** tab, in the **Personnel** field, click the ellipsis button.
 - d. In the **Universal Browser**, click **Personnel > People**, and then select your user name.
 - e. Click **OK**, or press `ENTER`.
 - f. Drag and drop a **Local Subprocess** into the workflow and double-click it, or double-click an existing subprocess.
 - g. From the subprocess configuration panel, click **Edit**.
 - h. In the **Name** field, enter a name, such as `WebFormStep`, and then click **OK**.
 - i. Click **Save**.
 - j. Click the **Overview** tab.
 - k. In the **Status** section, click **Enable**.
 8. Add a form to the workflow.
 - a. Click the **Workflow Editor** tab.
 - b. From the workspace, double-click the subprocess, `WebFormStep`.
 - c. From the **Standard Activities** panel, drag and drop a **Form** activity into the subprocess.
 - d. From the activity configuration panel, click **Edit**.
 - e. In the **Name** field, enter a name, such as `WebFormActivity`, and then click **OK**.
 - f. From the activity configuration panel, in **General Settings**, click the ellipsis button next to the **Forms** field.

The **Universal Browser** appears.

- g. In the **Universal Browser**, click **Global Displays > Forms**, and then select the form that you created, `My First Form`.
 - h. Click **OK**, or press `ENTER`.
 - i. Click **Save**.
9. Add activities to the subprocess to display form information to and request input from an operator. You can bind user input from the form's text box control (step 4) to workflow variables. For example, you can add a **Status Message** activity to your subprocess in order to deliver a message to your operator regarding a particular task, and request input.
- a. From the **Standard Activities** panel, drag and drop a **Status Message** activity into the subprocess after the **Form** activity.
 - b. From the activity configuration panel, in **General Settings**, click the drop-down arrow next to the **Message** field, and then select **Expression** to bind the message in your form.
 - c. In the **Enter Expression** field, enter the following expression: `"User entered:"+`
 - d. Click the **Workflow Variables** tab.
 - e. Expand the **Form** activity, `WebFormActivity`, that you added to the subprocess.
 - f. To complete the expression, double-click the text box value, `Text Box Control Name.Value`, that you created in step 4.
You have created the following expression: `"User entered:"+{WebFormActivity}.{Text Box Control Name.Value}`
 - g. Click **OK**.
 - h. From the **Standard Activities** panel, drag and drop a **Delay** activity into the subprocess after the **Status Message** activity.
 - i. In the **Timeout** field, select the **seconds** section, and then use the arrows to set a 5-second delay.
 - j. Click **Save**.
10. Add work instructions to the workflow.
- a. Click the **Work Instructions** tab.
 - b. From the **Task/Task Steps** list, select the task step, `WebFormStep`.

- c. In the **Work Instructions** area, enter instructions or insert an image.
 - d. Click **Save**.
11. Create a workflow schedule to start the workflow automatically, and then activate it so that it is available in production.
 - a. In the navigator, click **Workflow > Schedules**.
 - b. Click **Add Schedule**.
 - c. In the **Name** field, enter a name, such as `My Schedule`.
 - d. In the **Description** field, enter clarifying information about the resource, such as `This is my first schedule`.
 - e. Click **OK**, or press `ENTER`.
 - f. In the **Displays** panel, click **Workflow Management > Workflow Editor > Overview**.
 - g. In the **Select Workflow** section, click **Browse**.
The **Universal Browser** appears.
 - h. Select the workflow that you created in step 6, `My First Workflow`
 - i. In the **Start Task Filter** section, select the **Is schedule visible in Start Task window** check box.
 - j. Click **Save**.
 - k. In the **Status** section, click **Enable**.
12. Run and test the workflow.

- a. From the title bar, click the Task List icon,  .
- b. Click **Start Task**.
- c. Select the schedule that you created in step 11, `My Schedule`.
- d. Click **Start**.
The task that you created is now visible in the Task List. You can enter data into the form, view work instructions, and submit changes.

What to do next



Tip:

For Vision users, click the **TASKS** tab to see pending workflows.

Task List Filters Configuration

Users can perform context sensitive filtering in the Task List, which focuses a list of tasks or task steps on specific criteria.



Note:

Filter categories are used exclusively to filter tasks and task steps within the Task List. Categories that are associated with resources are used to group those resources and to allow you to search for and find those resources more effectively.

Administering Filters

You use the **Filter Editor** to manage filter configuration.

After adding a filter resource to the Workflow model, you can:

- associate the filter with personnel classes, users, computers and/or pieces of equipment; and
- specify predefined filter criteria.

Through a combination of assignments and criteria selection and configuration, you determine the conditions under which a filter is displayed in Task List, as well as the tasks or tasks steps that are displayed when an operator enables a filter.



Note:

To ensure that Task List users are able to work with **Predefined Filters** and **Personal Filters** during run time, you must assign permissions to those users. Some users, such as administrators, may have automatic permissions.

Filter Editor: Properties

The following table describes each available filter type.

Type	Value	Description
Task Name	User-defined	Filter by text that is contained in the name of a task as well as searches using a wildcard character (*).
Priority	User-defined	<p>Filter by the number that was assigned to the task when configuring the workflow, to indicate the task's priority for filtering purposes. The number(s) specified must be greater than zero. You can filter by multiple priority numbers by separating them with a comma, such as 1,2,3. You can also filter by a range of priority numbers, such as 1-5. These values are separated by a hyphen. If you want to filter by a series of ranges, you can separate them by commas (for example, 1-5, 10, 12-22).</p> <div data-bbox="1024 1213 1421 1436" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: You can enter whole numbers only, such as 0, 1, 2, 3, and so on. </div>
My Task Steps	N/A	<p>Filter by the current operator's assigned task steps.</p> <div data-bbox="1024 1570 1421 1793" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: All task steps that have been acquired by another operator are hidden. </div>

Type	Value	Description
Task Step State	N/A	<p>Filter by the current state of an operator's assigned task steps.</p> <p>Not Started</p> <p>Task step has not been reached in the execution cycle yet.</p> <p>Waiting for Start</p> <p>A manual start is required.</p> <p>Running</p> <p>In execution.</p> <p>Completed</p> <p>Completed as required.</p> <p>Cancelled</p> <p>Cancelled by the operator or system.</p> <p>Skipped</p> <p>No executed due to jump action or a condition branch.</p>
Categories	N/A	Filter tasks or task steps by a predefined grouping.
Location Assignment	N/A	Filters task steps by the equipment assigned to the task.
Personnel Assignment	NA	Filters task steps by the person assigned to the task.
Expiration	User-defined	Filters tasks or task steps that expire within a defined time range (from now until a defined time), or by a specific date and

Type	Value	Description
		<p>time. If a task has an expiry, then this filter applies to the task. If the task step only has an expiry, then this filter applies to the task step(s).</p> <div data-bbox="1024 514 1419 827" style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note: If a task or task step expires while the filter is on, the task or task step will continue to be displayed in the Task List.</p> </div>
Filtering Categories	N/A	Filters tasks and task steps by preconfigured categories to allow users to group and organize the Task List during run time.

Filter Editor: Assignments

The **Assignments** tab on the **Filter Editor** allows you to associate personnel classes, users, computers, and/or pieces of equipment with a given filter, as well as break such associations. Assignments determine the Task List users who have access to a given filter, such that the user must be among either the associated users or personnel classes; must be working on an associated computer; or must be working on a task or task step related to an associated piece of equipment. A filter can be applied to multiple resources, and a resource can have multiple filters applied to it.

Configure Task List filter assignments and criteria

After adding a filter resource to the Workflow model, you must configure the filter by associating it with the applicable personnel classes, users, computers, and/or pieces of equipment, and specifying the applicable predefined filter criteria. Through a combination of assignments and criteria selection and configuration, you determine the conditions under which a filter is displayed in Task List, as well as the tasks or tasks steps that are displayed when an operator enables a filter.

About this task**Note:**

When duplicating an equipment resource, any Task List filters that are assigned to the original equipment resource will not be automatically assigned to the new resource. Instead, you must manually assign the filter to the new equipment resource, as described beginning at step 6 of this procedure.

Procedure

1. In the navigator, click **Workflow > Filters**.
2. Select the filter that you want to configure.
3. In the **Displays** panel, click **Workflow Management > Filter Editor**.
4. Click the **Properties** tab.
5. Enable the applicable predefined filters by selecting one or more check boxes, and then perform the following steps, as applicable.
 - a. In the **Task Name** field, enter text related to the name of an existing task.
 - b. In the **Priority** field, enter a number (for example, 6), a range of numbers (for example, 1-5), or a series of ranges or numbers (for example, 1-5, 10, 12-22) that represents the number that was assigned to the task when configuring the workflow, to indicate the task's priority for filtering purposes.

**Note:**

You can enter whole numbers only, such as 0, 1, 2, 3, and so on.

- c. To view steps that apply to you only, select the **Only show my task steps** check box.
- d. Set the values for the **Task Step State** criterion by selecting from the corresponding drop-down list.

State	Description
Not Started	Task step has not been reached in the execution cycle yet.
Waiting for Start	A manual start is required.
Running	In execution.
Completed	Completed as required.

State	Description
Cancelled	Cancelled by the operator or system.
Skipped	Not executed due to a jump action or a condition branch.

- e. In the **Categories** field, select the check box, and then click **Set Categories** to select a preconfigured category.
- f. Set values for the **Assignments** criterion by selecting the related check box, and then:
 - For location, click **Set Location(s)** to select one or more resources.
 - For personnel, click **Set Personnel** to select a resource.
- g. Select the option applicable to the type of expiration filter that you want to configure, and then specify the related value.
 - Select **Expires In** to set how long from the current time the task or task step will expire.
 - Select **Expires Before** to set the date and time the task or task step is scheduled to expire.

6. Click the **Assignments** tab.

7. Click **Add**.

The **Universal Browser** appears.

8. Depending on the type of resource that you want to associate the filter with, select one of the following paths in the navigator.

For this type of resource...	Click...
Personnel classes	Personnel > Classes
Users	Personnel > People
Computers	Proficiency System > Proficiency System > Computers
Equipment	Equipment > Instances

- 9. Continue selecting from the lists that appear until you find the resource that you want to associate the filter with.
- 10. Click **OK**.
- 11. For each resource that you want to associate the filter with, repeats steps 7-10.
- 12. Click **Save**.

View and/or edit filter-resource assignments by Task List filter

After adding a filter resource to the Workflow model, you can use the **Filter Editor** display to view the resources currently associated with a filter, as well as associate additional resources with the filter and/or break the association between the filter and one or more resources.

Before you begin

You must already have added the filter resource to the Workflow model as well as added the related resources to personnel classes, equipment, users, and computers to perform this procedure.

Procedure

1. In the navigator, click **Workflow > Filters**.
2. Select the filter resource that you want to configure.
3. In the **Displays** panel, click **Workflow Management > Filter Editor**.
4. Click the **Assignments** tab.
The tab lists the resources currently associated with the filter.
5. From the **Assignments** tab, edit the resources associated with Task List the filter as required.

To...	Do this...
<p>Associate additional resources with the filter</p>	<ol style="list-style-type: none"> a. Click Add. <p>The Universal Browser appears.</p> <ol style="list-style-type: none"> b. From the navigator, select the model that contains the type of resource you want to associate with the filter, and then select a resource. c. Click OK. <p>The Assignments tab refreshes to display a listing for the newly associated resource.</p>
<p>Break the association between the Task List filter and one or more resources</p>	<ol style="list-style-type: none"> a. Select the resource listing. b. Click Remove. <p>The Assignments tab refreshes to reflect removal of the filter.</p>

Related information

[Configure Task List filter assignments and criteria \(on page 1322\)](#)

View and/or edit filter assignments by resource

After adding a filter resource to the Workflow model, you can use the **Filter Assignments** display to view the filters currently associated with a given resource, as well as associate additional filters with the resource and/or break the association between the resource and one or more filters.

Before you begin

You must already have added the filter resource to the Workflow model as well as added the related resources to personnel classes, equipment, users, and computers to perform this procedure.

Procedure

1. From the navigator, select the model that contains the resource you want to work with.
In the next pane, a list of resources appears.
2. Continue selecting from the lists that appear until you find the resource you want to work with.
3. In the **Displays** panel, click **Workflow Management > Filter Assignments**.
The **Filters** tab lists the filters currently associated with the resource.
4. From the **Filters** tab, edit the filters associated with the resource as required.

To...	Do this...
Associate additional filters with the resource	<ol style="list-style-type: none"> a. Click Add. <p>The Universal Browser appears.</p> <ol style="list-style-type: none"> b. In the navigator, click Workflow > Filters. c. Select the filter that you want to associate with the resource. d. Click OK. <p>The Filters tab refreshes to display a listing for the newly associated filter.</p>
Break the association between a resource and one or more filters	<ol style="list-style-type: none"> a. Select the filter listing. b. Click Remove. <p>The Filters tab refreshes to reflect removal of the filter.</p>

Related information

[Configure Task List filter assignments and criteria \(on page 1322\)](#)

Configure category filtering

You can filter the Task List by preconfigured categories. These categories organize concepts or features into groups. When you apply category filters, you focus information within your Task List.

Before you begin

You can add **Filtering Categories** to workflow tasks and/or subprocess/global subprocess task steps.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. **Optional:** Drag and drop a **Local Subprocess** into the workflow and click it, or click an existing subprocess.
6. Adding categories:

Click...	To add a category to a...
Task Step Details	Workflow
Task Details	Subprocess/Global Subprocess

7. In the **Filtering Categories** section, click **Assign**.
The **Univesal Browser** appears.
8. Select a category, and then click **OK**.
9. **Optional:** To remove a category, click **X**.

Task Assignments

Task assignments include routing workflow task to a location or personnel resource depending on the requirement of the workflow.

**Important:**

To validate the workflow, a location and/or personnel task assignment must be selected. However, if you are creating a workflow for a web-based environment, you should not assign a specific computer as a location.

General Panel Tab**Tip:**

You can use a Write Activity to change some property values at run time.

The following table describes the available options.

Attribute	Value	Description
Location	User-defined; Workflow level	Click the ellipsis button to select the default location that workflow task steps are routed to. <div data-bbox="1024 989 1419 1392" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Tip: Click the arrow to bind this location to a selected variable. Click the arrow to bind this location to a selected variable. You can bind this item to defined variables. </div>
Personnel	User-defined; Workflow level	Click the ellipsis button to select the default personnel that workflow task steps are assigned to. <div data-bbox="1024 1572 1419 1860" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Tip: Click the arrow to bind this location to a selected variable. Click the arrow to bind this location to a selected variable. </div>

Attribute	Value	Description
		 You can bind this item to defined variables.

Add or modify a location (equipment) for a workflow

Before you begin



Note:

When location is configured at the workflow level, the settings are applied to all subprocesses in the workflow unless they are overridden on the subprocess.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Click the ellipsis next to **Location**.
The **Universal Browser** appears.
6. Locate and select the resource you want to use, and then click **OK**.



Note:

Only one resource can be selected.

7. **Optional:** To change this value at run time, use a Write Activity.
8. If applicable, to perform a bind or reset, click the drop-down arrow.
9. Click **Save**.

Add or modify personnel for a workflow

Before you begin



Note:

When personnel are configured at the workflow level, the settings are applied to all subprocesses in the workflow unless they are overridden on the subprocess.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Click the ellipsis button next to **Personnel**.
The **Universal Browser** appears.
6. Locate and select the resource you want to use, and then click **OK**.

**Note:**

Only one resource can be selected.

7. **Optional:** To change this value at run time, use a Write Activity.
8. If applicable, to perform a bind or reset, click the drop-down arrow.
9. Click **Save**.

Continuously Looping Workflows

You can configure a workflow to run the preprocess through the postprocess logic in a continuous loop.

General Panel Tab**Tip:**

You can use a Write Activity to change some property values at run time.

The following table describes the available options.

Attribute	Value	Description
Continuous check box	User-defined; Workflow level	Select this check box to set the workflow to run in a continuous loop. For more information about continuous looping, see <i>Workflow Process Steps</i> .
Condition field	User-defined; Workflow level	If the Continuous check box is selected, you can enter an exit condition in this field to con-

Attribute	Value	Description
		Control the outcome of a continuous loop in a workflow. If no condition is entered, the workflow will not stop looping. If the condition is set to <i>False</i> , the looping stops, and the workflow's <i>Unload</i> step is processed. Removing continuous looping causes the workflow to fault.

Related reference

[Workflow Process Steps \(on page 1302\)](#)

Set a continuous workflow

You can configure a workflow to run the preprocess through the postprocess logic in a continuous loop.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. To set a continuous loop, select the **Continuous** check box.
6. **Optional:** To remove the continuous setting, clear the **Continuous** check box.
7. Click **Save**.

Set a condition to exit a continuously looping workflow

A condition is evaluated to a Boolean result. If the condition evaluates to **False**, the loop ends.

Procedure

1. In the **Condition** field, click **Edit Condition**.
The **Expression Editor** appears.
2. Enter an expression that evaluates to **False**.



Tip:

You can check the syntax and test your expression to ensure that it evaluates correctly.

3. Click **OK**.
4. **Optional:** To disable the expression and remove the loop, clear the **Continuous** check box.

**Note:**

The expression is not removed. As a result, it can be reinstated at a later time.

Task Name

Task Name is a general read property that is available at the workflow level.

**Tip:**

With these properties, you can build an expression or perform a bind.

Property	Value	Description
Task Name	User-defined; Workflow level	Specifies a unique name to distinguish it from the workflow name. <div data-bbox="1036 1016 1084 1071" data-label="Image"> </div> <div data-bbox="1094 1041 1174 1073" data-label="Section-Header">Note:</div> <div data-bbox="1094 1085 1386 1163" data-label="Text"> <p>This property can be used in a Write Activity.</p> </div>

Add a task name to a workflow

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource type list, select a resource.
3. In the *Displays* panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. From the workflow configuration panel, click the **Task Details** tab.
6. In the **Task Name** field, enter a name for the task.
7. If required, create an expression or perform a bind to set parameter values.
8. To remove the task name, click the arrow next the property you want to clear, and then select **Reset to Default**.
9. Click **Save**.

Highlight Color

Highlight Color is a general writable property that you can use to emphasize a task or task step during run time. When viewing these tasks during run time, you will observe a bar of color on the task that you set a color against.



Tip:

With this property, you can build an expression or perform a bind.

Property	Value	Description
Highlight Color	User-defined	<p>Optional. Specifies a color code for a task or task step to distinguish it from others. If a color is not set, then no highlight color is applied.</p> <div data-bbox="1036 905 1427 1520" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p>Tip: You can use this property to either:</p> <ul style="list-style-type: none"> • bind to another Color object or to a string that converts to a Color, or • use in a Write Activity to change the property at run time. </div>

Add custom colors to workflow tasks or subprocess task steps

You can add a custom color to a workflow or subprocess during configuration or during run time.

About this task

The following procedure contains more than one method for configuring color. Select the method that suits your run-time workflow requirements best.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the *Displays* panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Select the level you want to add a color to.

To add a color to a...	Perform the following action...
workflow task	Click the Task Details tab.
subprocess task step	Drag and drop a Local Subprocess into the workflow and click it, or click an existing subprocess. Click the Task Step Details tab.

6. To preconfigure a color, perform the following steps.
 - a. In the **Highlight Color** field, click the ellipsis button.
 - b. Go to step 7.
7. To preconfigure a color using a literal string value, perform the following steps.
 - a. At the workflow level, on the **General** tab, click **Edit** next to **Local Variables**.
The **Edit Local Variables** dialog box appears.
 - b. Click **Add**.
 - c. In the **Type** field, from the drop-down arrow, select **String**.
 - d. In the **Initial Value** field, enter a color value (for example, `red`), and then click **OK**.

**Note:**

You cannot use an irrelevant string value, such as **abcdefg**, or an invalid hex code.

- e. Select the level you want to add a color to.
 - For a workflow task, click the **Task Details** tab.
 - For a subprocess task step, select a subprocess, and then click the **Task Step Details** tab.
- f. In the **Highlight Color** field, select **Bind** from the drop-down arrow.
The **Select Variable** dialog box appears.

12. To remove the color, click the arrow next to the property you want to clear, and then select **Reset to Default**.
13. Click **Save**.

Configuring Electronic Signatures

To include a signature with a task/task step action or a form, add personnel with the authority to sign and verify forms. By default, all signatures are configured at the workflow level.

Override Workflow E-Signature Check Box (Subprocess Only)

All signatures are configured at the workflow level unless you override this setting at the subprocess level.

Attribute	Value	Description
Override Workflow E-Signature	User-defined	Select this check box to configure e-signature settings for a task step. When you select this check box, the Electronic Signature Settings area becomes active. For more information on configuring this section, see below.

Signature Required Check Box (Workflow and Subprocess)

If the task/task step action or form for your run-time workflow requires a signature, select the **Signature Required** check box.



Tip:

You can bind these properties to a resource related to a workflow or subprocess.

Attribute	Value	Description
Valid Performers	User-defined	Click the ellipsis button to select a personnel resource who has specific security authorization to perform this action. Members of this group signs forms related to the Form activity in the workflow.

Attribute	Value	Description
E-Signature Description	User-definedString	Specifies the reason for the signature.

Close if Signature Fails Check Box (Workflow and Subprocess)

If you select this check box, the task/task step action or form will close after the number of failure attempts has been reached. The workflow will continue to execute even if the signature does not work or is invalid.

Verifier Required Check Box (Workflow and Subprocess)

If the first signature (that is, the performer's signature) for your task/task step action or form requires a back-up signature to verify the original signature, then select this check box. If the form does not receive a valid signature, the workflow will not continue to execute.



Tip:

You can bind this property to a resource related to a workflow or subprocess.

Attribute	Value	Description
Valid Verifiers	User-defined	Click the ellipsis button to select a personnel resource who has specific security authorization to perform this action. Members of this group verify the signature(s) on the form after it is submitted.

Configure electronic signatures for use during workflow run time

You can configure electronic signatures to be available when a workflow is running to verify a task/task step action or a form.

Procedure

1. You can configure an electronic signature from various locations within a workflow depending on which action during run time you want to set an electronic signature against.

To add signature to a...	Do this...
Task	<ol style="list-style-type: none"> a. Open a workflow model resource. b. At the workflow level, click the E-Signature tab.
Task Step	<ol style="list-style-type: none"> a. Open a workflow model resource. b. Drag and drop a Local Subprocess into the workflow and select it, or select an existing subprocess. c. Click the E-Signature tab. d. Select the Override Workflow E-Signature check box.
Form	<i>Configure a Form activity.</i>

2. On the configuration panel, in the **Electronic Signature Settings** area, select the **Signature Required** check box.
3. In the **Valid Performers** field, click the ellipsis button to select the security group (personnel resource) that the signature performer must be a member of.
4. If you want the task/task step action or form to close and the workflow to proceed if the electronic signature fails, select the **Close if Signature Fails** check box.

**Note:**

A task/task step action or form can be cancelled without a signature if it has been preconfigured to do so; however, if you cancel, the task step is skipped and the workflow continues.

5. If a second signature is required before the task/task action or form can be submitted, do the following.
 - a. Select the **Verifier Required** check box.
 - b. In the **Valid Verifiers** field, click the ellipsis button to select the security group (personnel resource) that the signature verifier must be a member of.

6. In the **E-Signature Description** field, enter a description of the action being performed or the values that were entered on the task/task step action or form. This description is recorded with the electronic signature record.
7. Click **Save**.

Examples for Binding Valid Performers and Verifiers

Workflow provides the ability to bind valid performers and valid verifiers to an input parameter or the current value of a data item.

The following examples describe how to use binding for valid performers and verifiers.

I: Bind a valid performer or verifier to an input parameter on a global subprocess

You develop a global subprocess that will be used in multiple places in your facility. In this subprocess you have included a form activity that is configured with electronic signature requirements.

One of the workflows that uses the subprocess controls Line A and the other workflow controls Line B. You want to ensure that a different group of users can sign this form. For example, for Line A, the group Line A Operators must perform the signing; for Line B, the group Line B Operators must perform the signing.

To ensure that the correct group of users can sign the form, configure an input parameter for the subprocess so that the security group DirectoryResource data type is passed into the subprocess. On the form activity, bind the Valid Performers group property to the value of the input parameter.

In this way, you can specify different valid groups based on the workflow the subprocess is being used in without having to rewrite or duplicate your subprocess logic.

II: Bind a valid performer or verifier to the current value of a data item

To use this method, use a Search call method (see *IDirectorySearch* service interface) to dynamically get to a different security group and bind to its result.

**Note:**

If you follow either of these examples to configure a binding that uses a DirectoryResource data type, you must ensure that your logic is correctly set up so that it resolves to a security group at run time.

Automatic Task Selection

Auto Select is a read property that is available at the workflow level. You can set a task to be selected automatically in the Task List during run time by configuring client methods or workflow activities with the **Auto Select** feature.



Tip:

With these properties, you can build an expression or perform a bind.

Property	Value	Description
Auto Select	User-defined; Workflow level; Boolean	<p>If True, automatically selects a flagged task before other tasks. That is, the Task List itself determines which task to select when the current task is complete, when performing a manual refresh, or when first navigating to the Task List. If several tasks are flagged with auto select, they will be executed one after the other.</p> <p>You can also use <i>Client Methods</i> in a workflow to set this feature.</p> <div data-bbox="1036 1310 1084 1356" data-label="Image"> </div> <div data-bbox="1094 1318 1156 1354" data-label="Section-Header"> <p>Tip:</p> </div> <div data-bbox="1094 1360 1396 1488" data-label="Text"> <p>This property <u>cannot</u> be set at run time using a <i>Write Activity</i>.</p> </div>

Set a workflow to automatically start another workflow during execution

You can use Auto Select in a workflow with a Start Workflow or Start Schedule activity in order to use this activity to kick off another workflow and immediately select it.

Before you begin

You must have a second workflow or schedule preconfigured with Auto Select to be able to add it to a Start Workflow (or Start Schedule) activity.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. From the preconfigured workflow:
 - a. At the workflow level, click the **Task Details** tab in the configuration panel.
 - b. From the **Auto Select** drop-down list, select `True` to ensure that the task execution receives priority during run time.
 - c. If applicable, to perform a bind, click the drop-down arrow.
6. Select or insert a Local Subprocess.
7. To access activities, double-click the subprocess you want to work with.
8. Insert and configure a Start Workflow activity.

**Tip:**

You can also perform the same procedure with the Start Schedule activity.

9. On the *Settings* tab, in the **Workflow** field:
 - click the ellipsis button to select a workflow, or
 - from the drop-down arrow, select **Bind** to bind the workflow to a directory resource variable.

**Note:**

If you select **Bind**, you must select a template workflow.

10. **Optional:** In the **Template Workflow** field, click the ellipsis button to select a template workflow.
11. Click **Save**.

Select a parallel task step automatically during workflow execution

You can set a task or task step to be selected automatically in the Task List during run time by configuring workflow activities with the Auto Select feature.

About this task

You can use Auto Select in a workflow with parallel subprocesses in order to select a task step other than the first subprocess in the Parallel activity.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource type list, select a resource.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert a Parallel activity, and then add a Local Subprocess to each branch.
6. Configure each subprocess, and then select the second one to run this subprocess instead of the first.

**Tip:**

You can have more than two branches and set Auto Select on whichever subprocess meets your requirements.

7. From the subprocess configuration panel, click the **Task Step Details** tab.
8. From the **Auto Select** drop-down list, select `True` to ensure that the task execution receives priority during run time.
9. If required, create an expression or perform a bind to set parameter values.
10. Click **Save**.

Select a task automatically within a form during workflow execution using client methods

You can set a task or task step to be selected automatically in the Task List during run time by configuring client methods with the Set Selected Task feature.

Procedure

1. From the navigator, click **Global Displays > Forms**.
2. Add a form, or from the **Forms** list, select an existing form.
3. Click **Display Editor > Designer**.
4. Click the **Client Methods** tab.
5. Click **Add**.
A client method appears in the list.
6. Select the client method, and then click the **Properties** tab to configure the method.
7. In the **Source** field, click the ellipsis button.
The **Configure Method** dialog box appears.
8. Configure the properties, as required.

**Note:**

To view the full list of properties, select the **Advanced** check box.

9. Select one of the following client methods, and then click **Finish**.

- To automatically select a task at the workflow level, click **Set Selected Task**.
 - To automatically select a task step at the subprocess level, click **Set Selected Task Step**.
10. Based on the source selected, configure the following property.
 - If you set a task, in the **Task Name** field, enter the related name.
 - If you set a task step, in the **Task Step Name** field, enter the related name.
 11. If required, create an expression or perform a bind to set parameter values.
 12. Click **Save**.

Parameters

Parameters are used to pass data into and out of a workflow or subprocess, so that they can interact with other workflows and subprocesses.

There are two main types of parameters: input parameters and output parameters.



Tip:

You can use a Write Activity to change some property values at run time.

Parameter Attributes

The following table describes the available options.

Attribute	Value	Description
List	N/A	Displays the items created for the current workflow or subprocess.
Add	N/A	Click to add.
Remove	N/A	Click to delete the selection.
Name	User-defined	Specifies a display name for the selected property.
Type	User_defined	Specifies the type of data the parameter can store.
Direction	System-defined	Select either In for an input parameter, or Out for an output parameter. The direction indicates where you can read the value within the workflow or sub-

Attribute	Value	Description
		process. For example, you can only write to output parameters.
Description	User-defined	Provides more information that a user may find helpful.

Configure or modify a parameter for a workflow or subprocess

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. To configure a workflow, from the workflow configuration panel, click **Edit** next to the attributes collection you want to work with.
7. To configure a subprocess:
 - a. Double-click the subprocess you want to work with.
 - b. From the subprocess configuration panel, click **Edit** next to the attributes collection you want to work with.
8. **Optional:** To change this value at run time, use a Write Activity.
9. In the Edit dialog box, click **Add**.



Tip:

See section on each attribute for configuration.

10. From the **Parameters** list, select the parameter you want to work with.
11. From the **Details** list:
 - a. In the **Name** field, enter a name.
 - b. From the **Type** list, select a data type.
 - c. From the **Direction** list, select either **In** (for input) or **Out** (for output).
 - d. Optionally, in the **Description** field, enter more information for this attribute.
12. **Optional:** To change this value at run time, use a Write Activity.
13. Click **OK**, and then click **Save**.

Local Variables

Local variables are used as local data sources that are available to the running workflow during the lifetime of the workflow instance.

Local variables can be written to by an activity bind or from a Write activity. For example, you have a workflow with a subprocess called Subprocess1, and the subprocess contains two activities: Activity1 and Activity2. Subprocess1 also contains a local variable called Variable1. When the workflow runs and reaches Subprocess1, Activity1 writes to Variable1. Activity2 can now read from Variable1, and Variable1 can be manipulated depending on the type of activity that Activity2 is defined as.

Local variables can be defined at both the workflow and subprocess levels. Subprocess local variables can be accessed only from within the specific subprocess they were created in, whereas workflow level local variables can be accessed from anywhere within the workflow. For example, an activity in one subprocess can write to a workflow local variable, and in turn be accessed again in another subprocess.



Tip:

You can use a Write Activity to change some property values at run time.

Local Variable Properties

The following table describes the available options.

Attribute	Value	Description
List	N/A	Displays the items created for the current workflow or subprocess.
Add	N/A	Click to add.
Remove	N/A	Click to delete the selection.
Name	User-defined	Specifies a display name for the selected property.
Type	User-defined	Specifies the data type for the variable.
Initial Value	User-defined	Specifies a value that corresponds with the data type specifications.

Attribute	Value	Description
Allow tracking for this variable	N/A	<p>Select this check box to add the local variable to a trackable variables list in <i>Tracking Configuration</i>, and to ensure that data related to a task is available when a running workflow completes.</p> <div data-bbox="1027 569 1419 877" style="border: 1px solid #f0e68c; border-radius: 10px; padding: 10px; background-color: #fff9c4;"> <p> Important: To track a variable, you must also select it in Trackable Variables in the Tracking Configuration tab.</p> </div> <div data-bbox="1027 909 1419 1266" style="border: 1px solid #4db6ac; border-radius: 10px; padding: 10px; background-color: #e0f2f1;"> <p> Note: To view a tracked variable after a workflow has run, enable tracking at both the process (workflow) and subprocess level.</p> </div>
Description	User-defined	Provides more information that a user may find helpful.

Related reference

[Tracking Data in Workflows \(on page 1573\)](#)

Related information

[Configure or modify a local variable for a workflow or subprocess \(on page 1346\)](#)

Configure or modify a local variable for a workflow or subprocess

About this task

Attributes include:

- parameters
- local variables
- data items
- external events
- namespaces

Procedure

1. Add a local variable attribute:
 - a. In the Navigator, click **Workflow**.
 - b. In the Folders list, select the resource type you want to work with.
 - c. In the resource type list, select the resource you want to work with.
 - d. In the Show Compatible panel, click Workflow Editor.
 - e. Click the Workflow Editor tab.
 - f. To configure a workflow: From the workflow configuration panel, click Edit next to the attributes collection you want to work with.
 - g. To configure a subprocess, double-click the subprocess you want to work with. Then, from the subprocess configuration panel, click Edit next to the attributes collection you want to work with.
 - h. Optionally, to change this value at run time, use a Write activity.
 - i. In the Edit dialog box, click **Add**.



Tip:

See section on each attribute for configuration.

2. From the Local Variables list, select the variable you want to work with.
3. From the Details list:
 - a. In the Name field, enter a name.
 - b. From the Type list, select a data type.
 - c. In the Initial Value field, enter a value that corresponds to the data type.
 - d. If applicable, select or clear the **Allow tracking for this variable** check box.

e. Optionally, in the Description field, enter more information for this attribute.



Note:

To view a tracked variable after a workflow has run, enable tracking at both the process (workflow) and subprocess level.

4. To change this value at run time, use a Write Activity.
5. Click OK, and then click **Save**.

Data Items

Data items are data bound to data sources that are external to the current workflow. If the data item has a defined data type, then the data item acts as a local variable of that data type.

For example, you can create a data item bound to the value of an iFIX tag. Each time the data item is accessed, the current value of the iFIX tag is returned. You can use this data item in an expression to determine a branching condition, which uses external data to drive execution of the workflow.



Note:

You can configure up to 50 data items for each workflow definition.

Parameterized Data Items

A parameterized data item allows you to select a template data source (a preconfigured resource) both at the workflow level and at the subprocess level. A parameterized data item is different because you can select a different data source at run time. You can use parameterized data items when you configure schedules, run a test workflow, or invoke a Start Workflow activity.

You can specify the run-time data source data items in both Workflow Schedules and in the Workflow Editor **Test** tab. The data item configured in the workflow is used as a template to validate that the one chosen in the schedule is the correct data type.

In order for a parameterized data item to be effective, a run-time data source must be in a workflow. The parameterized data item on a subprocess cannot be directly linked to a data source; it must be bound to a workflow data item. This also means that it cannot be bound to a parameterized data item defined on another subprocess. If the parameterized data items are either unbound or are bound to an incompatible workflow data item, your workflow will not be valid.

Binding Data Items

Data items can be used directly in activity binds and expressions; that is, you can bind activity properties to a data source through a data item when the data item has a defined type. They can also be used as the target or source of a Write activity. For specific activities such as Read Data Items or Write Data Items, data items can be used as the target of these activities.

Data items are also used indirectly. For example, when you use multiple columns of data, or when advanced configuration of the read or write is required, you must use a Read Data Items or Write Data Items activity to access the data item.

Data items can also be used in Create Event activity expressions by binding an event variable to a data item. For more information, see *Conditional Events*.

Data Item Attributes



Note:

Values displayed in the activity configuration panel are the default values of the selected data item. If you change the values in this configuration panel when configuring an activity, then these new values will be used.

The following table describes the available options.

Attribute	Value	Description
List	N/A	Displays the items created for the current workflow or sub-process.
Add	N/A	Click to add.
Remove	N/A	Click to delete the selection.
Name	User-defined	Specifies a display name for the selected property.
Data Source	System-defined	Click the ellipsis button to open the Configure Data Source dialog box to select the specific data source you want this data item to bind to.

Attribute	Value	Description
Make this data item a parameter	System-defined	Select this check box, to configure the data source in Workflow Schedules and the Workflow Editor Test tab.
Type	System-defined	Displays the data type of the selected data source.
Readable	System-defined	Indicates that the data source property can be read. This information displays after you have added a data source.
Writeable	System-defined	Indicates that the data source property can be written to. This information displays after you have added a data source.
Valid Event Variable	System-defined	Indicates that the data item can be used as an event variable. This information displays after you have added a data source. For more information, see <i>Condition Event ExpressionVariables</i> .
Description	User-defined	Provides more information that a user may find helpful.

Configure or modify a data item for a workflow or subprocess

About this task

Attributes include:

- parameters
- local variables
- data items
- external events
- namespaces

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. To configure a workflow, from the workflow configuration panel, click **Edit** next to the attributes collection you want to work with.
7. To configure a subprocess:
 - a. Double-click the subprocess you want to work with.
 - b. From the subprocess configuration panel, click **Edit** next to the attributes collection you want to work with.
8. **Optional:** To change this value at run time, use a Write Activity.
9. In the Edit dialog box, click **Add**.



Tip:

See section on each attribute for configuration.

10. From the **Data Items** list, select the item you want to work with.
11. From the Details list:
 - a. In the **Name** field, enter a name.
 - b. In the **Data Source** field, click the ellipsis button, and then select and configure the required data item.
See *Configure a data source item for a resource*.
 - c. If applicable, select or clear the **Make this data item a parameter** check box.
 - d. Optionally, in the **Description** field, enter more information for this attribute.
12. Click **OK**, and then click **Save**.

Turn data items into parameters

Before you begin

**Note:**

Parameterized data items defined on a subprocess must be bound to data items defined on a workflow when the subprocess is added to a workflow.

Attributes include:

- parameters
- local variables
- data items
- external events
- namespaces

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. To configure a workflow, from the workflow configuration panel, click **Edit** next to the attributes collection you want to work with.
7. To configure a subprocess:
 - a. Double-click the subprocess you want to work with.
 - b. From the subprocess configuration panel, click **Edit** next to the attributes collection you want to work with.
8. **Optional:** To change this value at run time, use a Write Activity.
9. In the Edit dialog box, click **Add**.

**Tip:**

See section on each attribute for configuration.

10. From the **Data Items** list, select the item you want to work with.

11. From the Details list, select the **Make this data item a parameter** check box.
12. Click **OK**, and then click **Save**.

Related information

[Add a resource to the workflow model \(on page 1395\)](#)

External Events

External events allow the workflow or subprocess to use events defined outside of the workflow.

These external events that a subprocess or workflow can bind to can be objects that are events within the application, or from other software applications that have published events to Workflow. Some of the external events available include: production, conditional, timed, and work order events.

Properties

The following table describes the available options.

Attribute	Value	Description
List	N/A	Displays the items created for the current workflow or subprocess.
Add	N/A	Click to add.
Remove	N/A	Click to delete the selection.
Name	User-defined	Specifies a display name for the selected property.
External Event	User-defined	Click the ellipsis button to open the Configure Event dialog box to select the event you want to bind the current workflow or subprocess to.
Description	User-defined	Provides more information that a user may find helpful.

Related reference

[Create Event Activity \(on page 1488\)](#)

[Event Handlers for Web-based Forms \(on page 1519\)](#)

Related information

[Configure or modify an external event for a workflow or subprocess \(on page 1354\)](#)

[Configure a Create Event activity \(on page 1489\)](#)

[Add an event handler to a Form activity \(on page 724\)](#)

[Configure an event handler \(on page 1518\)](#)

Configure or modify an external event for a workflow or subprocess

Procedure

1. Add an external event attribute. See *Add attributes to a workflow or subprocess*.
2. From the **External Events** list, select the item you want to work with.
3. From the **Details** list:
 - a. In the **Name** field, enter a name.
 - b. In the **External Event** field, click the ellipsis button, and then select and configure the required event.
 - c. Optionally, in the **Description** field, enter more information for this attribute.
4. Click **OK**, and then click **Save**.

Namespaces

A namespace is a logical design-time naming convenience, used mainly to define scope in an application and organize classes and other types in a single hierarchical structure.



Note:

Code activities use the imported namespaces on the workflow and subprocess to resolve simple type names.

The namespace concept is part of the .NET Framework, and is used to organize types into logical groupings. A type's full name is made up of a namespace and a simple name. Although a type's full name can always be used in a Code activity, it is more common to use the type's simple name.

Example

System.dita.ditaDocument is a type that is used to work with a document containing XML data. The namespace for that type is System.dita and the simple type name is XmlDocument. The full name, System.dita.ditaDocument, can be used directly in a Code activity to refer to the type. However, if the System.dita namespace is added to the Namespaces list, the simple name, XmlDocument, can be used instead which increases the clarity of the code. Using the namespace also means that the simple names for other types in the System.dita namespace can also be used.

Namespace Browser

The following table describes the available options.

Attribute	Value	Description
Available Namespaces	Microsoft; System	Accessible namespaces from within Workflow.
Imported Namespaces	Microsoft.VisualBasic; System	A type's simple name can be used in namespaces that have been added to this list.

Add a namespace reference to a workflow or subprocess

Prerequisites

[Add a resource to the workflow model \(on page 1395\)](#)

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. To configure a workflow, from the workflow configuration panel, click **Edit** next to the attributes collection you want to work with.
7. To configure a subprocess:
 - a. Double-click the subprocess you want to work with.
 - b. From the subprocess configuration panel, click **Edit** next to the attributes collection you want to work with.
8. In the **Available Namespaces** list, select the namespace to import.

**Tip:**

Navigate through the menus by expanding the parent namespace.

9. Click the right arrow to add the selected namespace to the **Imported Namespaces** list.
10. Click **OK**, and then click **Save**.

Remove a namespace reference from a workflow or subprocess

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. To configure a workflow, from the workflow configuration panel, click **Edit** next to the attributes collection you want to work with.
7. To configure a subprocess:
 - a. Double-click the subprocess you want to work with.
 - b. From the subprocess configuration panel, click **Edit** next to the attributes collection you want to work with.
8. In the **Imported Namespaces** list, select the namespace you want to remove.
9. Click the left arrow to remove the namespace from the **Imported Namespaces** list.
10. Click **OK**, and then click **Save**.

Linked Documents

You can link documents or web sites in workflow tasks, subprocess task steps, and schedule tasks. If the document is defined at the task level, it is visible in all task steps.

In the Linked Documents area, you can attach a document or web site address for the person performing a running workflow task (or a subprocess task step) to reference. These documents can include anything that an operator requires to perform or complete an operation.

Linked documents appear in the Documents panel in the Task List and Task Management while a workflow is in a run-time state. They are embedded in this panel unless your Microsoft Windows configuration does not allow this, and can be in any application format that the user's workstation

supports. However, if a document opens outside the linked documents panel, the user is responsible for closing it.



Tip:

When you link a document at the workflow level, that is, to a task, it remains open while the task runs (unless the user closes it or opens another document). All embedded documents close automatically when a task is completed.

Restrictions for Linked Documents

Mobile Task List

In order to access a linked document on a mobile device, you must host the file on a web server. As a result, network file paths and non-web-based resources are inaccessible.

Access Permissions



Note:

You must have the ability to apply or assign documents to a task.

Example for Linking Documents

Example

If a maintenance operator is performing a maintenance task on a piece of equipment that he has not worked on for some time, the service manual can be an important reference. The service manual can be linked and then viewed for reference while carrying out the maintenance work.

User Interface for Linking Documents

The following table describes the document linking options:



Note:

The Display Name and URL/UNC properties allow you to build an expression or perform a bind.

Property	Value	Description
List	User-defined	Displays a list of documents that are linked to the selected item.

Property	Value	Description
		They can be arranged by document name.
Add	User-defined	Click this button to add a linked document to the current item.
Preview	N/A	Click to view the document or web site you inserted.
Remove	User-defined	Click this button to remove a linked document from the current item.
Name	User-defined	<p>Specifies a unique identifier for the document you are linking.</p> <div data-bbox="1027 856 1424 1031" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Tip: This property is used only in a Write activity.</p> </div>
URL/UNC Text	User-defined	<p>Specifies the name of the document you are linking, which is displayed in the Task List and/or Task Management during runtime.</p> <div data-bbox="1027 1304 1424 1661" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Tip: The file name of any browsed document you attach is automatically entered in this field. You can accept this file name or enter a different name.</p> </div>
URL/UNC	User-defined	Displays the path to the document that you are linking. You can enter the full path, or select a document from a directory. You

Property	Value	Description
		<p>must always enter an absolute path.</p> <p>The following documents types are supported in Safari: .doc,- .docx, .xls, .xlsx, .ppt, .pptx, .pdf,- .jpg, .png, .gif</p> <div data-bbox="1024 583 1422 758" style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px;">  Note: For URLs, you must enter <code>http://.</code> </div>
Visible	System-defined	<p>Indicates whether a document is available in the Task List. You can also set a bind or create an expression to alter the visibility of the document at run time.</p> <div data-bbox="1024 1031 1422 1747" style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px;">  Note: Setting the Visible property to <code>False</code> does not close an open linked document. The document must be closed manually by the user. You can use this feature to configure documents in advance, and then allow the logic of the workflow to determine which document is required and used at run time. </div>

Remove a workflow linked document

Prerequisites

[Add a resource to the workflow model \(on page 1395\)](#)

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. From the workflow configuration panel, click the **Task Details** tab.
7. In the **Linked Documents** list, click the **Remove** button.
8. Click **Save**.

Add and configure workflow linked documents

Before you begin



Tip:

- Mobile-sized Task List only: To view Microsoft Office 2007 and later documents inside an Internet Explorer browser window, add a registry patch.
- Web Task List only: To open embedded documents, set the URI for each link to an absolute path, save documents in a folder on a server, and give the folder share permissions.

Prerequisites

[Add a resource to the workflow model \(on page 1395\)](#)

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. From the workflow configuration panel, click the **Task Details** tab.
7. In the **Linked Documents** area, click **Add**.
8. In the **Name** field, enter a unique name for the file or web site you are attaching.

The default entry is `LinkedDocument1`, and so on.

9. In the **URL/UNC Text** field, enter the text that will be displayed to the user when the task is running.
10. In the **URL/UNC** field:
 - Click the ellipsis button to open the Browse window to attach a document.
 - Enter the absolute path or full web site address for the document you are attaching.
11. In the **Visible** field, click the first arrow, and then select either **True** or **False**.
12. If required, create an expression or perform a bind to set parameter values.
13. To view the document or web site, click **Preview**.
14. Click **Save**.

Related information

[Add a registry patch \(on page 1395\)](#)

Work Instructions

The **Work Instructions** tab is where you can input detailed instructions for a person performing a task or task step.

These instructions appear as a **Work Instructions** button in the Task List. If no information is available, this section is not visible to an operator. The information can include anything that an operator requires to perform a workflow task or a subprocess task step, or finds useful to complete the task/task step.



Note:

If content in the work instructions area exceeds a certain threshold, the editor may not perform optimally during design time, as well as in the Task List during run time. To prevent this issue, use the following recommendations: use small graphic files; use small amounts of text, or several line breaks, within the body of the work instructions area.

Updating Work Instructions During Run Time

You can use keywords to replace text in work instructions and email subjects and bodies. This text is replaced with keyword values in a workflow at run time.

For more information, see *Replace keyword text during run time*.

Attributes

The following table describes the available options.

Attribute	Value	Description
Task/Task Steps list	N/A	Specifies the name of the currently selected workflow, and the names of all subprocesses associated with that workflow.
Work Instructions	User-defined	Select the workflow or a subprocess, and then enter instructions related to that task or task step.

Add work instructions to a workflow or subprocess

The **Work Instructions** tab is where you can input detailed instructions for a person performing a task or task step.

Prerequisites

[Add a resource to the workflow model \(on page 1395\)](#)

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Work Instructions** tab.
6. From the **Task/Task Steps** list, to enter instructions for a:
 - task, click the workflow title.
 - task step, select a subprocess.
7. In the **Work Instructions** area, enter instructions or insert an image.



Tip:

You can type text or insert information copied from another application, as well as graphics.

8. Click **Save**.

What to do next



Note:

To update work instructions during run time, see the instructions related to keyword replacement.

Related reference

[Keyword Properties \(on page 1496\)](#)

[Update Work Instruction Keywords Activity \(on page 1495\)](#)

[Work Instructions \(on page 1361\)](#)

[Replace Keywords Activity \(on page 1495\)](#)

Related information

[Replace keyword text during run time \(on page 1494\)](#)

Modify or delete workflow or subprocess work instructions

Before you begin



Note:

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Work Instructions** tab.
6. From the **Task/Task Steps** list, to enter instructions for a:
 - task, click the workflow title.
 - task step, select a subprocess.
7. Modify or delete the instructions, as required.
8. Click **Save**.

Related reference

[Keyword Properties \(on page 1496\)](#)

[Update Work Instruction Keywords Activity \(on page 1495\)](#)

[Work Instructions \(on page 1361\)](#)

Related information

[Replace keyword text during run time \(on page 1494\)](#)

[Add work instructions to a workflow or subprocess \(on page 1362\)](#)

[Add a resource to the workflow model \(on page 1395\)](#)

Configuring Operator Task List Actions

Operators can perform a variety of actions with a task or task step, such as assigning priority, entering expiration reasons, cancelling, performing jumps, acquiring and releasing the task, and reassigning task steps.

Available Actions

When a workflow is running in the Task List, operators can perform various advanced actions within a *task* and/or *task step*. Many of these actions require specific user permissions that can be set in the Key Sets section of Workflow. You can also configure some of these actions to require an electronic signature for verification.

These actions are available for a *task*:

- Set a priority (task only)
- Cancel a task
- Enter an expiry comment

These actions are available for a *task step*:

- Acquire a task step
- Release a task step
- Cancel a task step
- Enter an expiry comment
- Jump to a different task step
- Reassign a task step

These actions are available for both a *task* and a *task step*:

- Cancel a task step
- Enter an expiry comment

Historical Task Information

When a workflow is finished running, the related task information is available in **Task History**. Operators can enter expiry comments on completed or expired tasks and task steps.

Set the priority of a workflow task

You can apply a numeric value for a task. This priority setting does not affect the execution of the workflow; it is used only for sorting and filtering within the Task List.

Prerequisites

[Add a resource to the workflow model \(on page 1395\)](#)

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. From the workflow configuration panel, click the **Task Details** tab.
6. In the **Priority** field, enter a numeric value.



Note:

You can enter whole numbers only, such as 0, 1, 2, 3, and so on.

7. If required, create an expression or perform a bind to set parameter values.
8. Click **Save**.

Related reference

[Task Priority \(on page 1366\)](#)

Set a workflow task to expire

You can set an expiration date and/or time for workflow tasks when a workflow is running.

Prerequisites

[Add a resource to the workflow model \(on page 1395\)](#)

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. From the workflow configuration panel, click the **Task Details** tab.
6. From the **Expiration Type** drop-down list, select:
 - **Relative** to set a countdown
 - **Absolute** to set an exact expiry time
7. For a relative expiration:
 - a. In the **Duration** field, set the number of days, hours, minutes, or seconds until the expiration.
 - b. To set a conditional event, select the **Condition Event** check box.
 - c. Click **Configure Event**.
8. For an absolute expiration:
 - a. In the **Duration** field, enter a date and time.
 - b. Click **OK**.
9. If required, create an expression or perform a bind to set parameter values.
10. Click **Save**.

Related reference

[Expiration in Schedules \(on page 1551\)](#)

[Task Expiration \(on page 1367\)](#)

Related information

[Set a subprocess task step to expire \(on page 1393\)](#)

[Override an expiration for a schedule \(on page 1553\)](#)

Task Priority

Task Priority is a general read property that is available at the workflow level. Task Priority is a general read property that is available at the workflow level. You can apply a numeric value for a task. This priority setting does not affect the execution of the workflow; it is used only for sorting and filtering within the Task List.

Restrictions

A priority number is set in a running workflow at the task level only.

Access Permissions



Note:

Operators must have authorization to perform certain actions in the Task List, which can be assigned to them in the **Security Editor**. They may, however, still view the contents in this area.

Task Expiration

Operators can enter an explanation as to why a task or task step expired.



Note:

Operators must have authorization to perform certain actions in the Task List, which can be assigned to them in the **Security Editor**. They may, however, still view the contents in this area.

An expiration is a visual indication about how much time is available to perform a workflow task (or a subprocess task step). Expirations determine the duration of a task or task step when a workflow or a subprocess is in a run-time state.

In a workflow instance, if a task or task step expires, you can still complete it. The entire process is not lost; however, the task or task step is logged as having timed out, and the workflow state changes and initiates an event that you can hook into. This event can also be used as a means of escalation. If required, you can then run the task or task step again. A running task is not interrupted if a task expires.

Example

During the execution of a task, one of the task steps expires because of an equipment maintenance issue. The task supervisor can, during the execution of the subsequent task steps, go back and complete or restart the expired task step when the employee and equipment are available to complete the task step.

Expiration Options



Note:

The Absolute and Relative expiration types allow you to build an expression or perform a bind.

Action	Value	Description
Name	System-defined; Default selection	No expiration is set for the selected item.

Action	Value	Description
Absolute	User-defined	Select this option to define an exact date and time when this item expires. In the Duration field, select a date and time.
Relative	User-defined	Select this option to provide a countdown timer to indicate how long this item has to complete. In the Duration field, you can set the number of days, hours, minutes, or seconds for this item to expire. <div data-bbox="1024 764 1419 1079" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Tip: Insert a condition event to: provide the completion countdown, or ensure a condition is met to expire this item.</p> </div>
Relative: Condition Event	User-defined	Select this check box to configure a condition event for the item's relative time. Click the Configure Event link to configure the conditional event. For information, see <i>Condition Events</i> .

Canceling Tasks

Operators can cancel an entire task or an individual task step in the Task List.



Note:

Operators must have authorization to perform certain actions in the Task List, which can be assigned to them in the **Security Editor**. They may, however, still view the contents in this area.

Continuous Loop

If a task is cancelled during the execution of one loop in a continuous workflow, a new task is created for the next loop. The current cancelled task is removed from the Task List and is replaced by the new one, starting again at the first step.

Tracking in a Loop

If tracking in a workflow is turned on, each task appears as two separate tasks in **Task History**.

Acquiring and Releasing Tasks

Task steps may be acquired by an operator if it has not already been acquired by another operator. Then, if required, an operator may release a task step so that it can be acquired by another operator.



Note:

Operators must have authorization to perform certain actions in the Task List, which can be assigned to them in the **Security Editor**. They may, however, still view the contents in this area.

Setting Task Steps to be Acquired

Tasks can appear in more than one Task List. For this reason, workflows are automatically set up to ensure that the first request made that acquires a task step takes ownership of that task step until it is either completed or released. If two or more operators try to acquire a task at the same time, only one is able to start the task. Subsequent personnel receive a message stating that another operator has already acquired it. If more than one user is performing a preparatory task at the same time, such as filling in a form, the first operator to submit the form receives acknowledgment for having completed the task. In some circumstances, the task step is acquired as soon as data is entered in a form.

Task steps can be acquired in several ways, such as:

- using the Jump, Cancel, or Manual Start features;
- submitting or cancelling a form;
- programmatically acquiring the step using the client method in the Forms Designer (not available in the mobile task lists); or
- marking the subprocess to automatically acquire the task step, so that the first operator whose task list is populated with the task step will acquire it.

Task Jumps

Operators can skip from one task step to another in the Task List when a workflow is running.

**Note:**

Operators must have authorization to perform certain actions in the Task List, which can be assigned to them in the **Security Editor**. They may, however, still view the contents in this area.

How *Jump* Affects Running Workflows

The *Jump* feature affects the execution of workflows in the following ways:

- In If/Else activities at the workflow level, the *jump* does not re-evaluate the condition if these activities were skipped. The condition is evaluated again if the If/Else activity is executed again; that is, if you jump to the start of the If/Else.
- When jumping back to a task step, the new one appears in the Task List, and the old one is removed; however, all task steps display in the **Task History** tab, if applicable.
- The workflow structure that created the task determines which steps are valid to jump to at a particular stage of task execution.

Reassigning Tasks Steps

In an active, running workflow, operators can reassign a task step to a different person, piece of equipment, or location, as required.

**Note:**

Operators must have authorization to perform certain actions in the Task List, which can be assigned to them in the **Security Editor**. They may, however, still view the contents in this area.

Task Step Configuration

Task Step (Subprocess) Configuration

A subprocess is part of a user-defined process in a workflow.

For a workflow to run, all subprocesses must be valid (contain no errors) and enabled (active).

A workflow must be constructed of one or more subprocess. Each subprocess is a self-contained entity where the majority of the logic is added to run a workflow. Subprocesses group a set of related activities in a workflow and appear as a single task step. A subprocess can exist only between the presubprocess and postsubprocess.

Subprocesses allow you to create a series of steps to accomplish a task, assignment, event-driven, or activated process. The contents of a subprocess consist of activities that can be placed in any order

depending on the requirement. You can think of the contents of a subprocess as a workflow without the defined structure and in a simple linear path.

Within a subprocess activity, you can select a manual start option for a workflow. This option allows you to configure a subprocess to be either an automated aspect of a workflow, or to require operator interaction within the workflow process. When the manual start check box is cleared, the subprocess runs automatically when the workflow is activated. When this check box is selected, the workflow stops when a manual start subprocess is reached within the workflow. At this point, an operator is required to intervene in the process and complete an action before the workflow can continue.

Within a workflow definition, you can select a continuous option for the workflow instance. However, in a continuous loop, you cannot loop a single subprocess. During the continuous loop, every step from the preprocess to the postprocess is completed in the continuous loop cycle. You can, for example, put in conditional activities to facilitate a small loop inside of the subprocess in case a step is not completed or an error found.

Inputs and outputs of a subprocess can be linked from and to other subprocesses or execution cycle steps. For example, you can create a dependency on the output of subprocess1 for subprocess2, or the input of subprocess1 is dependent on the output of the preprocess or load steps. Therefore, you can ensure that at each step of the execution cycle, there are dependencies that must be met before the cycle can continue onto the next step.

Subprocesses also have a built-in completion code variable that is always tracked. While a subprocess is running, this variable can be set to any available completion code for a project. A default value must be supplied for this completion code.

**Note:**

You can configure up to 1000 subprocesses.

Properties for Subprocesses

You can use properties to further define the resource you are configuring.

These properties are scoped differently. If the property does not explicitly have a value, the value will be the default value for the type. See *Scope Definitions*.

Properties

The following table describes the available options.

Property	Read	Write	Description
Auto Select	All	None	Flags a task to be executed prior to non-flagged task steps in a subprocess during run time.
Absolute Expiry Time	All	All	In a preprocess or a pre-subprocess, you can set an absolute expiry time (DateTime value) if the Expiry Type is set to Absolute.
Relative Expiry Time	All	All	In a preprocess or a pre-subprocess, you can set a relative expiry time (TimeSpan value) if the Expiry Type is set to Relative.
Expiry Type	All	All	Allows you to set the subprocess's expiration to Absolute, Relative, or None during run time.
Assigned Location	All	All	This value indicates the assigned location of a computer and/or a piece of equipment.
Assigned Personnel	All	All	This value indicates the personnel assigned to a task step.
Completed	All	None	Read Only Allows you to determine if a specific executed subprocess or task step

Property	Read	Write	Description
			<p>has completed or been cancelled.</p> <p>This value is set to False for all subprocesses at the beginning of the process loop. It is set to True after the subprocess activity execution finishes if the task step was completed (not cancelled). It is reset to False if a jump operation to or before the subprocess is performed.</p> <p>If an unhandled fault occurs during subprocess execution, this flag is set to False.</p>
Completion Code	All	None	Read Only
Current Computer	All	None	Read Only
Current User	All	None	Read Only
Enabled	All	External	True if enabled, False if not.
Executed	All	None	<p>Read Only</p> <p>Allows you to determine if a specific subprocess or task step has been executed by the workflow engine.</p> <p>The value for this property is set to False for all subprocesses by the</p>

Property	Read	Write	Description
			Process Workflow activity before it executes the preprocess. The value is set to True after each subprocess finishes execution, but before the postsubprocess executes. The value is not set to True if the subprocess is skipped.
Global Subprocess	All	None	Read Only
Id	All	None	Read Only Unique key in the scope of a workflow/schedule definition that identifies a step within the workflow/schedule.
Is Global Subprocess	All	None	Read Only
Is Acquired	All	All	True if subprocess task step has been acquired by another user, False if it has not. Validation: can only be set to False at runtime when task step is already acquired.
Manual Start	All	None	True if the task step needs to be manually started, False if it does not.
My Task List Documents State	All	All	Type: MyTaskListExpanderLayout

Property	Read	Write	Description
			Possible values: User-Configuration, Auto, Expanded, Collapsed
My Task List Documents View	All	All	Type: DocumentsView Possible values: Input, WorkInstructions, LinkedDocuments
My Task List Documents Visibility	All	All	True if the Task List's documents are visible, False if they are not.
My Task List Expander Position	All	All	Type: ExpanderPosition Possible values: Horizontal and Vertical
My Task List Input State	All	All	Type: MyTaskListExpanderLayout Possible values: User-Configuration, Auto, Expanded, Collapsed
Task Step Instance Name	All	All	The string name of the task step instance.
My Task List Input Visibility	All	All	N/A
Name	All	None	Read Only Display name of the step within the workflow/schedule.
Subprocess Group	All	None	Process Scope
Task Step Instance Id	All	None	Read Only Process Scope

Property	Read	Write	Description
			Unique key that identifies a task step (instance). Task step instance identifiers are assigned each time a subprocess execution begins; that is, just before the presubprocess executes. This property states undefined if you attempt to access the task step instance ID for a task step when it has not started.
Task Step Instance Name	All	None	Read Only Process Scope String name of the task step instance.
Task Step Instance Expired	All	None	Read Only Process Scope Time that the task instance can no longer be performed.
Visible	All	None	True if the task step is visible, False if it is not.

Scope Definitions

The scope of a property, as well as the location of the activity that it is defined on in a workflow, determines if it can be referenced or accessed by a workflow bind.

The following table defines the type of properties that are accessible in a workflow:

Scope Type	Properties Accessible	Properties Not Accessible
Workflow-scoped	Load, Preprocess, Presubprocess, Postsubprocess, Postprocess, Unload, all local and global subprocesses, subprocess flow controls, and the workflow itself	None
Process-scoped	Preprocess, Presubprocess, Postsubprocess, Postprocess, all local and global subprocesses, subprocess flow controls	Load, Unload
Subprocess-scoped	Presubprocess, Postsubprocess, and the subprocess itself	Load, Preprocess, Postprocess, Unload, all local and global subprocesses, subprocess flow controls, the workflow

Limits for Subprocesses

There are restrictions to the elements that comprise a workflow. These limits ensure a quality functioning workflow.

You can create a maximum of 1000 subprocesses.



Important:

Data item limits include child activities.

Limit	Restriction	Result
Data items and External events (including contained activities)	100	Validation error.

Subprocess Configuration Panel-Within the Subprocess

When you are inside a subprocess, you can configure attributes that are specific to the subprocess. You can access the internal subprocess from the Workflows folder, or from the Subprocesses folder, in the Workflow model.

**Tip:**

You can use a Write Activity to change some property values at run time.

Inputs and Outputs

As the subprocess author, you define the attributes and values of the input and output parameters. There are no system-defined attributes.

Panel Attributes

The following table describes the available options.

Attribute	Value	Description
View	User-defined	Allows you to choose between two views: <ul style="list-style-type: none"> • Activity sequence (default view) • Fault handlers: select to view or configure fault handlers
Default Completion Code	User-defined	Click the ellipsis to add a preconfigured completion code.
	User-defined	Click the Edit link to configure.
	User-defined	Click the Edit link to configure.
	User-defined	Click the Edit link to configure.
	User-defined	Click the Edit link to configure.
	User-defined	Click the Edit link to configure.
	User-defined	Click the Edit link to configure.

Configure subprocesses

Procedure

1. In the navigator, click **Workflow**.
2. From the Folders list:

Click...	To access a...
Subprocesses	global subprocess
Workflows	subprocess that is already part of a workflow

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Double-click the subprocess you want to work with.
6. From the subprocess configuration panel, click **Edit** next to the attributes collection you want to work with.

Enable Property

Enabled is a read/write property that can be read from and written to at run time.

This property is defined on a subprocess and is process-scoped. This property can be configured using a literal value, a bind, or an expression. It can also be modified at run time using either a `or` or `.` When Enabled is set to True, the activity or subprocess is included in an activated/running workflow or schedule.



Note:

The global subprocess must be enabled in order to execute at run time.

When Enabled is set to False, the activity or subprocess does not execute; however, the presubprocess and postsubprocess steps run, as is typical in process procedures. When Enabled is set to **False**, the task step does not appear in the task client display.



Note:

You can select this option only from within a subprocess or activity and you cannot set the Enabled property for an activity during run time.

Property	Value	Details
Enabled	User-defined	Click the first arrow to set: <ul style="list-style-type: none"> • True: Subprocess executes. • False: Subprocess does not execute. The task step is skipped.

Property	Value	Details
		<div data-bbox="1024 268 1419 485" style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px; margin-bottom: 10px;">  Note: The global subprocess must be enabled in order to execute at run time. </div> <p data-bbox="1024 520 1333 548">Click the second arrow to:</p> <ul data-bbox="1084 600 1414 1079" style="list-style-type: none"> <li data-bbox="1084 600 1414 716">• Reset to Default: Subprocess is reset to True. <li data-bbox="1084 737 1414 898">• Bind: Opens the Select Variable dialog box where you can select an applicable property or parameter. <li data-bbox="1084 919 1414 1079">• Expressions: Opens the Expression Editor where you can create an expression. <p data-bbox="1024 1129 1409 1205">With this property, you can build an expression or perform a bind.</p>

Configure a subprocess to run within a workflow

Before you begin



Note:

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.

5. Click the **Workflow Editor** tab.
6. Drag and drop a **Local Subprocess** into the workflow and double-click it, or double-click an existing subprocess.

**Note:**

If required, configure the workflow.

7. From the subprocess configuration panel, click the **General** tab.
8. In the **Enabled** field, click the first arrow, and then select either **True** or **False**.

**Note:**

The global subprocess must be enabled in order to execute at run time. For more information, see *Enable and disable a subprocess*.

9. If required, create an expression or perform a bind to set parameter values.
10. Click **Save**.

Manual Start Property

When the **Manual Start** property is set to **True**, the subprocess requires operator interaction to begin execution.

Manual Start is defined on a subprocess. It can be configured using a literal value.

**Note:**

If **Manual Start** is set to **True**, you cannot set Visible to **False**. That is, when **Manual Start** is set to **True**, the activity appears in the task display.

Property	Value	Details
Manual Start	User-defined	Click the first arrow to set: <ul style="list-style-type: none"> • True: Requires operator interaction. • False: Subprocess executes automatically.

Property	Value	Details
		Click the second arrow to: <ul style="list-style-type: none"> Reset to Default: Subprocess is reset to True.

Set a subprocess to Manual Start

Before you begin



Note:

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Drag and drop a **Local Subprocess** into the workflow and double-click it, or double-click an existing subprocess.



Note:

If required, configure the workflow.

6. From the subprocess configuration panel, click the **General** tab.
7. In the **Manual Start** field, click the first arrow, and then select **True**.
8. Click **Save**.

Visible Property

When the Visible property is set, the corresponding task step will appear in the Task List during run time.

Visible is a read-only property that can be read and used as a bind source, but cannot be written to. It is also defined on a subprocess and is process-scoped. This boolean property can be configured using a literal value. If Visible is set to **False**, then a corresponding step does not appear in the Task List or Task Management; however, the subprocess is still executed.

**Important:**

- If this property is set to **True**, then the personnel and/or location assignments in the workflow configuration panel must be set.
- If the subprocess has any activities or actions that require user interaction, this property must be set to **True**.

Examples of user interaction include:

- Task expiry
- Work instructions
- Linked documents
- Form activity
- Status Message activity
- Manual Start property

Property	Value	Details
Visible	User-defined	<p>Click the first arrow to set:</p> <ul style="list-style-type: none"> • True: Displays the subprocess in Task List/Task Management. • False: Hides the subprocess in Task List/Task Management. <p>Click the second arrow to:</p> <ul style="list-style-type: none"> • Reset to Default: Subprocess is reset to True.

Make a subprocess task step visible during run time

Before you begin



Note:

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. Drag and drop a **Local Subprocess** into the workflow and double-click it, or double-click an existing subprocess.



Note:

If required, configure the workflow.

7. From the subprocess configuration panel, click the **General** tab.
8. In the **Visible** field, click the first arrow, and then select either **True** or **False**.



Tip:

If Visible is set to True, you must also configure Task Assignments. For more information, see *Configuring workflows*.

9. Click **Save**.

Subprocess Group Property

You can use the Subprocess Group property to treat similar subprocesses as a group.

It is a read-only string property that can be read and used as a bind source, but cannot be written to. It is also defined on a subprocess and is workflow-scoped. This property can be configured using a literal value.

Property	Value	Details
Subprocess Group	User and system-defined	<p>Enter a label for the subprocess group (literal string).</p> <p>Click the arrow to:</p> <ul style="list-style-type: none"> Reset to Default: Subprocess is reset to an empty string.

Label a subprocess group

Before you begin



Note:

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. Drag and drop a **Local Subprocess** into the workflow and double-click it, or double-click an existing subprocess.



Note:

If required, configure the workflow.

7. From the subprocess configuration panel, click the **General** tab.
8. In the **Subprocess Group** field, enter a label for the group.



Note:

If this is a global subprocess, a default label is already entered, which you can override.

9. Click **Save**.

Task Step Assignments

Task steps assignments include routing the task steps of a subprocess (or portions of a workflow) to a location or personnel resource depending on the requirement of the workflow.

These task steps can be sent to an assigned computer or an operator at that station to complete. However, if a task step is specified to a location only, any user will be able to see that task step at that location. For example, a single workflow may have a task step assignment that requires that an item be scanned with a bar code scanner at station 2, while station 3 has a task step assignment that is a request for a count of items.



Important:

In order for a workflow to run, it must be assigned to a person or a piece of equipment.

Override Task Assignment Settings

By default, subprocess assignments are configured at the workflow level, where you can bind personnel and location properties, as well as write to the assignments from anywhere in the workflow. However, you can override these assignments for a specific subprocess and reassign the tasks to other personnel or locations.

Reassignments

Task steps can be reassigned during run time. This can happen in one of the following ways:

- From Task List/Task Management, you can intervene and reassign the task. When steps are reassigned, they can no longer be acquired and any predefined acquirement is removed.
- A workflow can change at run time using a Write Activity.
- A workflow can make a change to another workflow through the Start Schedule or Start Workflow activities.



Note:

If a workflow is in a continuous loop, task step reassignments do not affect the next loop.

Properties

Property	Value	Description
Override Workflow Task Assignments	User-defined	Select this check box to allow a user to override the equipment

Property	Value	Description
		and/or personnel resources assigned to the subprocess activity.
Location	User-defined	<p>When the override check box is selected, click the ellipsis button to select an alternate location to route the task step of the current subprocess to.</p> <div data-bbox="1027 621 1419 890" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;"> <p> Tip: Click the arrow to bind this location to a selected variable or reset it to the default value.</p> </div>
Personnel	User-defined	<p>When the override check box is selected, click the ellipsis button to select an alternate personnel resource to route the task step of the current subprocess to.</p> <div data-bbox="1027 1161 1419 1430" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;"> <p> Tip: Click the arrow to bind a personnel resource to a selected variable or reset it to the default value.</p> </div>

Add or modify equipment (location) for a subprocess

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.

6. Drag and drop a **Local Subprocess** into the workflow and double-click it, or double-click an existing subprocess.



Note:

If required, configure the workflow.

7. To configure a subprocess:
 - a. From the subprocess configuration panel, click **Override Workflow Task Assignments**.
 - b. Click the ellipsis button next to Location.
The **Universal Browser** appears.
8. Locate and select the resource you want to use, and then click **OK**.



Note:

Only one resource can be selected.

9. **Optional:** To change this value at run time, use a Write Activity.
10. If required, create an expression or perform a bind to set parameter values.
11. Click **Save**.

Add or modify personnel for a subprocess

Before you begin



Note:

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. Drag and drop a **Local Subprocess** into the workflow and double-click it, or double-click an existing subprocess.

**Note:**

If required, configure the workflow.

7. To configure a subprocess:
 - a. From the subprocess configuration panel, click **Override Workflow Task Assignments**.
 - b. Click the ellipsis button next to Personnel.
The **Universal Browser** appears.
8. Locate and select the resource you want to use, and then click **OK**.

**Note:**

Only one resource can be selected.

9. **Optional:** To change this value at run time, use a Write Activity.
10. If required, create an expression or perform a bind to set parameter values.
11. Click **Save**.

Acquire Task Steps

You can set task steps to be automatically acquired by another person or machine during the execution of a workflow.

Property	Value	Description
Automatically acquire task step	User-defined	Select this check box to set the computer and/or user settings to the computer and/or user on the first Task List display that it appears on. <div data-bbox="1036 1570 1086 1621" data-label="Image"></div> Note: When steps are reassigned, they can no longer be acquired and any predefined acquirement is removed.

Set a task step to be automatically acquired

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. Drag and drop a **Local Subprocess** into the workflow and double-click it, or double-click an existing subprocess.

**Note:**

If required, configure the workflow.

7. From the subprocess configuration panel, click the **General** tab.
8. Select the **Automatically acquire task step** check box.
9. To remove this setting, clear the **Automatically acquire task step** check box.
10. Click **Save**.

Enable and disable a subprocess

Before you begin

**Note:**

To edit a subprocess, it must be disabled.

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. On the **Overview** tab:

Click...	To...
Enable	activate the subprocess
Disable	deactivate the subprocess

**Note:**

The Enabled property of the subprocess activity within a workflow must also be set to True for the subprocess to execute at run time. For more information, see *Configure a subprocess to run within a workflow*.

Task Step Configuration

Under the **Task Step Details** tab in the **Subprocess Configuration Panel**, you can input or attach more information related to a workflow that helps a user perform task steps in the Task List and/or Task Management.

General Panel Tab

Using these settings, you can correlate your models and processes with relevant task step and subprocess data that allows you to perform queries using reporting. Some properties also allow you to affect how a workflow interacts with the Task List and make run time decisions about the execution of subprocesses.

**Important:**

If operator interaction of a task step is required, the *Visible Property* must be set to **True**.

Task Step Name

Task Step Name is a general read property that is available at the subprocess level.

**Tip:**

With these properties, you can build an expression or perform a bind.

Property	Value	Description
Task Step Name	User-defined; Subprocess level	Specifies a unique name to distinguish it from the subprocess

Property	Value	Description
		<p>name. No two subprocesses in one workflow should have the exact same name.</p> <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Tip: This property can be used in a Write Activity.</p> </div>

Add a task step name to a subprocess

Before you begin



Note:

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. Drag and drop a **Local Subprocess** into the workflow and double-click it, or double-click an existing subprocess.



Note:

If required, configure the workflow.

7. From the subprocess configuration panel, click the **Task Step Details** tab.
8. In the **Task Step Name** field, enter a name for the task.
9. If required, create an expression or perform a bind to set parameter values.
10. To remove the task step name, click the arrow next the property you want to clear, and then select **Reset to Default**.
11. Click **Save**.

Set a subprocess task step to expire

You can set an expiration date and/or time for subprocess task steps when a workflow is running.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. Drag and drop a **Local Subprocess** into the workflow and double-click it, or double-click an existing subprocess.



Note:

If required, configure the workflow.

7. From the subprocess configuration panel, click the **Task Step Details** tab.
8. From the **Expiration Type** drop-down list, select:
 - Relative to set a countdown
 - Absolute to set an exact expiry time
9. For a relative expiration:
 - a. In the **Duration** field, set the number of days, hours, minutes, or seconds until the expiration.
 - b. To set a conditional event, select the **Condition Event** check box.
 - c. Click **Configure Event**.
10. For an absolute expiration:
 - a. In the **Duration** field, enter a date and time.
 - b. Click **OK**.
11. If required, create an expression or perform a bind to set parameter values.
12. Click **Save**.

Add and configure subprocess linked documents

Before you begin**Tip:**

- Mobile-sized Task List only: To view Microsoft Office 2007 and later documents inside an Internet Explorer browser window, add a registry patch.
- Web Task List only: To open embedded documents, set the URI for each link to an absolute path, save documents in a folder on a server, and give the folder share permissions.

**Note:**

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. From the subprocess configuration panel, click the **Task Step Details** tab.
6. In the **Linked Documents** area, click **Add**.
7. In the **Name** field, enter a unique name for the file or web site you are attaching.
The default entry is `LinkedDocument1`, and so on.
8. In the **URL/UNC Text** field, enter the text that will be displayed to the user when the task is running
9. In the **URL/UNC** field:
 - Click the ellipsis button to open the Browse window to attach a document.
 - Enter the absolute path or full web site address for the document you are attaching.
10. In the **Visible** field, click the first arrow, and then select either **True** or **False**.
11. If required, create an expression or perform a bind to set parameter values.
12. To view the document or web site, click **Preview**.
13. Click **Save**.

Remove a subprocess linked document

Before you begin**Note:**

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. Drag and drop a **Local Subprocess** into the workflow and double-click it, or double-click an existing subprocess.

**Note:**

If required, configure the workflow.

7. From the subprocess configuration panel, click the **Task Step Details** tab.
8. In the **Linked Documents** list, click the **Remove** button.
9. Click **Save**.

Add a resource to the workflow model

To configure resources and to develop a workflow, you must first add resources to the workflow model.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. Click **Add**.
4. In the **Name** and **Description** fields, enter a name and description for the resource.
5. Click **OK**.
6. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
7. Click the **Workflow Editor** tab.
8. Configure the resource as required.

Add a registry patch

About this task

The Mobile-sized Task List supports opening files inside browser windows. However, with Microsoft Office 2007 and later, some applications (such as Word, Excel, and PowerPoint) open only outside the browser window regardless of settings. Therefore, opening these file types using the Mobile-sized Task List does not work.

To open these files inside a browser window using the Mobile-sized Task List, apply the following registry patch. These registry settings must also be on the machine that the mobile client is deployed to.

Procedure

1. Open a text editor.
2. Copy and then paste the following text into the editor:

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Word.Document.8]
"BrowserFlags"=dword:80000024

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Word.RTF.8]
"BrowserFlags"=dword:80000024

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Word.Document.12]
"BrowserFlags"=dword:80000024

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Word.DocumentMacroEnabled.12]
"BrowserFlags"=dword:80000024

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Excel.Sheet.8]
"BrowserFlags"=dword:80000A00

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Excel.Sheet.12]
"BrowserFlags"=dword:80000A00

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Excel.SheetMacroEnabled.12]
"BrowserFlags"=dword:80000A00

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\Excel.SheetBinaryMacroEnabled.12]
"BrowserFlags"=dword:80000A00
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\PowerPoint.Show.8]

"BrowserFlags"=dword:80000A0

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\PowerPoint.Show.12]

"BrowserFlags"=dword:80000A0

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\PowerPoint.ShowMacroEnabled.12]

"BrowserFlags"=dword:80000A0

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\PowerPoint.SlideShow.8]

"BrowserFlags"=dword:80000A0

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\PowerPoint.SlideShow.12]

"BrowserFlags"=dword:80000A0

[HKEY_LOCAL_MACHINE\SOFTWARE\Classes\PowerPoint.SlideShowMacroEnabled.12]

"BrowserFlags"=dword:80000A0

[HKEY_CURRENT_USER\Software\Microsoft\Windows\Shell\AttachmentExecute\{0002DF01-0000-0000-C000-000000000046}]

"Word.Document.12"=hex:

"Word.Document.8"=hex:

"Excel.Sheet.8"=hex:

"Excel.Sheet.12"=hex:

"PowerPoint.Show.8"=hex:

"PowerPoint.Show.12"=hex:
```

3. Save the file as `FileName.reg`.
4. Double-click `FileName.reg` to run this registry script.
5. Restart the Mobile-sized Task List.
6. Start Microsoft Internet Explorer.

Highlight Color

Highlight Color is a general writable property that you can use to emphasize a task or task step during run time. When viewing these tasks during run time, you will observe a bar of color on the task that you set a color against.

**Tip:**

With this property, you can build an expression or perform a bind.

Property	Value	Description
Highlight Color	User-defined	<p>Optional. Specifies a color code for a task or task step to distinguish it from others. If a color is not set, then no highlight color is applied.</p> <div data-bbox="1023 682 1421 1291" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <div data-bbox="1039 703 1088 745" data-label="Image"></div> <div data-bbox="1096 709 1156 745" data-label="Section-Header">Tip:</div> <div data-bbox="1096 751 1412 831" data-label="Text"> <p>You can use this property to either:</p> </div> <div data-bbox="1161 882 1412 1228" data-label="List-Group"> <ul style="list-style-type: none"> • bind to another Color object or to a string that converts to a Color, or • use in a Write Activity to change the property at run time. </div> </div>

Add custom colors to workflow tasks or subprocess task steps

You can add a custom color to a workflow or subprocess during configuration or during run time.

About this task

The following procedure contains more than one method for configuring color. Select the method that suits your run-time workflow requirements best.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the *Displays* panel, click **Workflow Management > Workflow Editor**.

4. Click the **Workflow Editor** tab.
5. Select the level you want to add a color to.

To add a color to a...	Perform the following action...
workflow task	Click the Task Details tab.
subprocess task step	Drag and drop a Local Subprocess into the workflow and click it, or click an existing subprocess. Click the Task Step Details tab.

6. To preconfigure a color, perform the following steps.
 - a. In the **Highlight Color** field, click the ellipsis button.
 - b. Go to step 7.
7. To preconfigure a color using a literal string value, perform the following steps.
 - a. At the workflow level, on the **General** tab, click **Edit** next to **Local Variables**. The **Edit Local Variables** dialog box appears.
 - b. Click **Add**.
 - c. In the **Type** field, from the drop-down arrow, select **String**.
 - d. In the **Initial Value** field, enter a color value (for example, `red`), and then click **OK**.

**Note:**

You cannot use an irrelevant string value, such as **abcdefg**, or an invalid hex code.

- e. Select the level you want to add a color to.
 - For a workflow task, click the **Task Details** tab.
 - For a subprocess task step, select a subprocess, and then click the **Task Step Details** tab.
 - f. In the **Highlight Color** field, select **Bind** from the drop-down arrow. The **Select Variable** dialog box appears.
 - g. From the Workflow Local Variables properties, select the string value that you entered earlier, and then click **OK**.
 - h. Go to step 10.
8. To set the highlight color to another property of type **Color**, perform the following steps.

If you want to add the color from one subprocess to another subprocess, you require more than one subprocess in the workflow.

- a. Select the level you want to add a color to.
 - For a workflow task, click the **Task Details** tab.
 - For a subprocess task step, select a subprocess, and then click the **Task Step Details** tab.
 - b. In the **Highlight Color** field, select **Bind** from the drop-down arrow.
The **Select Variable** dialog box appears.
 - c. From the properties of the workflow or subprocess that contains a configured color value, select **Highlight Color**, and then click **OK**.
 - d. Go to step 10.
9. To set a color using a Write activity, perform the following steps.
- a. Double-click a subprocess, insert a Write activity, and then click **Add Write**.
The **Select Variable** dialog box appears.
 - b. Select **Highlight Color**, and then click **OK**.
 - a. In the **Value** field, click the ellipsis button.
 - b. Got to step 7.
10. To create a custom color from the color palette, perform the following steps.
- a. In the **Highlight Color** field, click the ellipsis button.
The **Color** palette window appears.
 - b. To create a custom color, perform one of the following actions.
 - In the **Basic colors** section, click a color, and then click **OK**.
 - In the color panel, select a custom color, click **Add to Custom Colors**, and then click **OK**.
11. If required, from the drop-down arrow, select **Bind** to bind to another Object type or String property.
12. To remove the color, click the arrow next to the property you want to clear, and then select **Reset to Default**.
13. Click **Save**.

Configuring Electronic Signatures

To include a signature with a task/task step action or a form, add personnel with the authority to sign and verify forms. By default, all signatures are configured at the workflow level.

Override Workflow E-Signature Check Box (Subprocess Only)

All signatures are configured at the workflow level unless you override this setting at the subprocess level.

Attribute	Value	Description
Override Workflow E-Signature	User-defined	Select this check box to configure e-signature settings for a task step. When you select this check box, the Electronic Signature Settings area becomes active. For more information on configuring this section, see below.

Signature Required Check Box (Workflow and Subprocess)

If the task/task step action or form for your run-time workflow requires a signature, select the **Signature Required** check box.



Tip:

You can bind these properties to a resource related to a workflow or subprocess.

Attribute	Value	Description
Valid Performers	User-defined	Click the ellipsis button to select a personnel resource who has specific security authorization to perform this action. Members of this group signs forms related to the Form activity in the workflow.
E-Signature Description	User-definedString	Specifies the reason for the signature.

Close if Signature Fails Check Box (Workflow and Subprocess)

If you select this check box, the task/task step action or form will close after the number of failure attempts has been reached. The workflow will continue to execute even if the signature does not work or is invalid.

Verifier Required Check Box (Workflow and Subprocess)

If the first signature (that is, the performer's signature) for your task/task step action or form requires a back-up signature to verify the original signature, then select this check box. If the form does not receive a valid signature, the workflow will not continue to execute.



Tip:

You can bind this property to a resource related to a workflow or subprocess.

Attribute	Value	Description
Valid Verifiers	User-defined	Click the ellipsis button to select a personnel resource who has specific security authorization to perform this action. Members of this group verify the signature(s) on the form after it is submitted.

Configure electronic signatures for use during workflow run time

You can configure electronic signatures to be available when a workflow is running to verify a task/task step action or a form.

Procedure

1. You can configure an electronic signature from various locations within a workflow depending on which action during run time you want to set an electronic signature against.

To add signature to a...	Do this...
Task	<ol style="list-style-type: none"> a. Open a workflow model resource. b. At the workflow level, click the E-Signature tab.

To add signature to a...	Do this...
Task Step	<ol style="list-style-type: none"> a. Open a workflow model resource. b. Drag and drop a Local Subprocess into the workflow and select it, or select an existing subprocess. c. Click the E-Signature tab. d. Select the Override Workflow E-Signature check box.
Form	<i>Configure a Form activity.</i>

2. On the configuration panel, in the **Electronic Signature Settings** area, select the **Signature Required** check box.
3. In the **Valid Performers** field, click the ellipsis button to select the security group (personnel resource) that the signature performer must be a member of.
4. If you want the task/task step action or form to close and the workflow to proceed if the electronic signature fails, select the **Close if Signature Fails** check box.

**Note:**

A task/task step action or form can be cancelled without a signature if it has been preconfigured to do so; however, if you cancel, the task step is skipped and the workflow continues.

5. If a second signature is required before the task/task action or form can be submitted, do the following.
 - a. Select the **Verifier Required** check box.
 - b. In the **Valid Verifiers** field, click the ellipsis button to select the security group (personnel resource) that the signature verifier must be a member of.
6. In the **E-Signature Description** field, enter a description of the action being performed or the values that were entered on the task/task step action or form. This description is recorded with the electronic signature record.
7. Click **Save**.

Automatic Task Step Selection

Auto Select is a read property that is available at the subprocess level. You can set a task step to be selected automatically in the Task List during run time by configuring client methods or workflow activities with the Auto Select feature.

**Tip:**

With these properties, you can build an expression or perform a bind.

Property	Value	Description
Auto Select	User-defined; Boolean	<p>If True, automatically selects a flagged task step and selects it before other task steps. That is, the Task List itself determines which task step to select when the current task step is complete, when performing a manual refresh, or when first navigating to the Task List. If several task steps are flagged with auto select, they will be executed one after the other.</p> <p>You can also use <i>Client Methods</i> in a workflow to set this feature.</p> <div data-bbox="1023 1113 1421 1333" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Tip: This property cannot be set at run time using a Write Activity. </div>

Set a workflow to automatically start another workflow during execution

You can use Auto Select in a workflow with a Start Workflow or Start Schedule activity in order to use this activity to kick off another workflow and immediately select it.

Before you begin

You must have a second workflow or schedule preconfigured with Auto Select to be able to add it to a Start Workflow (or Start Schedule) activity.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource type list, select a resource.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. From the preconfigured workflow:
 - a. At the workflow level, click the **Task Details** tab in the configuration panel.
 - b. From the **Auto Select** drop-down list, select `True` to ensure that the task execution receives priority during run time.
 - c. If applicable, to perform a bind, click the drop-down arrow.
6. Select or insert a Local Subprocess.
7. To access activities, double-click the subprocess you want to work with.
8. Insert and configure a Start Workflow activity.

**Tip:**

You can also perform the same procedure with the Start Schedule activity.

9. On the *Settings* tab, in the **Workflow** field:
 - click the ellipsis button to select a workflow, or
 - from the drop-down arrow, select **Bind** to bind the workflow to a directory resource variable.

**Note:**

If you select **Bind**, you must select a template workflow.

10. **Optional:** In the **Template Workflow** field, click the ellipsis button to select a template workflow.
11. Click **Save**.

Select a parallel task step automatically during workflow execution

You can set a task or task step to be selected automatically in the Task List during run time by configuring workflow activities with the Auto Select feature.

About this task

You can use Auto Select in a workflow with parallel subprocesses in order to select a task step other than the first subprocess in the Parallel activity.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert a Parallel activity, and then add a Local Subprocess to each branch.

6. Configure each subprocess, and then select the second one to run this subprocess instead of the first.

**Tip:**

You can have more than two branches and set Auto Select on whichever subprocess meets your requirements.

7. From the subprocess configuration panel, click the **Task Step Details** tab.
8. From the **Auto Select** drop-down list, select `True` to ensure that the task execution receives priority during run time.
9. If required, create an expression or perform a bind to set parameter values.
10. Click **Save**.

Interface Layout Options for the Task List

Task Step Views allows you to preconfigure the interface visible to operators when a Task List is running.

**Tip:**

You can use a Write Activity to change some property values at run time.

You can also configure other visible elements of the Task List and copy these configurations to other computers to ensure consistency across all workstations. For more information, see *User Interface Controls in the Task List*.

Preferred Views for Task Steps

You can set a default view when an operator selects a *task step* in the **Views** panel of the Task List. For example, if you select the **Linked Documents** radio button, the content in this section is visible immediately to the operator when he selects the **Views** panel.

**Note:**

If the view that you have set does not have content, the **Views** panel defaults to the **Forms** view.

Properties	Value	Description
Forms	System-defined	Appears as the Forms button in the Task List.

Properties	Value	Description
Linked Documents	System-defined	Appears as the Linked Documents button in the Task List. If content is available in a <i>task</i> or <i>task step</i> , when an operator clicks this button, a document opens.
Work Instructions	System-defined	Appears as the Work Instructions button in the Task List. If content is available <i>task</i> or <i>task step</i> , when an operator clicks this button, information relevant to performing a task appears.

Navigate to Task Step Views

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. Double-click the subprocess you want to work with.
7. From the subprocess configuration panel, click **Edit** next to **Task Step Views**.

The Task Step Views dialog box appears.

Set the Task List panel layout

Task Step Views allows you to preconfigure the interface visible to operators when a Task List is running.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.

5. Click the **Workflow Editor** tab.
6. Double-click the subprocess you want to work with.
7. From the subprocess configuration panel, click **Edit** next to **Task Step Views**.
The *Task Step Views* dialog box appears.
8. To set the default content in the **Views** panel, select one of the options below.
 - Linked Documents
 - Work Instructions
 - Forms
9. **Optional:** To change this value at run time, use a Write Activity.
10. Click **OK**.

User Interface Controls in the Task List

My Task List Editor allows you to select user controls that you want to be visible in the Task List, and copy these configurations to other computers or personnel classes.

Controls



Note:

This functionality is not applicable to the Mobile-sized Task List.

All controls of the Task List are visible unless configured otherwise. If the check box next to a control is cleared, the control will not be available during run time.



Tip:

All changes to the My Task List Editor section must be saved before you can copy the configuration to other computers or personnel classes.

The following table describes the available options.

Controls	Value	Description
Task Title	User-defined	Displays the given name for a workflow task.
Task Grid	User-defined	Provides an overview of tasks, including title, duration, expiry, and priority.

Controls	Value	Description
Task Context Navigator	User-defined	Provides the ability in real time to view and navigate all tasks currently running.
Task Step Listing	User-defined	Provides a visual display of the task steps being executed or that require execution.
Task Step Context Navigator	User-defined	Provides the ability in real time to view and navigate all tasks steps currently running.
Status Message	User-defined	Allows for display of a message that provides further information to the operator regarding the currently running task step.
Release Task Steps Dialog	User-defined	Allows unauthorized users to release task steps during run time. Clear this check box to prevent task step release by unauthorized users.
Task/Task History	User-defined	Displays the Tasks and Task History tabs in the control bar. Clear this check box to prevent navigation from current to past tasks during run time.
Task List Context	User-defined	Shows all information related to computer, user, and equipment context. Clearing this check box hides this information along with the Context button, and results in display of the preconfigured context for the task.
Equipment Context Editing	User-defined	Shows the Context button. Clear this check box to hide the button and allow operators to only view

Controls	Value	Description
		equipment context information during run time, not modify it.
Views	User-defined	Allows for interaction with manual inputs during run time, such as filling in a form, performing a manual start for a task, signing a document, or viewing a document. If no manual input for a workflow is required, then this button is not required.
Details	User-defined	Provides for viewing general read-only information, such as start/end times and personnel/location assignments, about a selected task or task step.
Actions	User-defined	Provides for performing advanced actions such as cancelling, acquiring, and reassigning a task; entering comments; and jumping to other task steps. If this button is not available, these operations cannot be performed.
Filters	User-defined	Allows for task filtering by specific criteria such as task name, priority, state, assignment, and expiration.
Start Task	User-defined	Provides for starting a schedule that is associated with a task or task step during run time.
Step Layout	User-defined	Allows for configuring the view of the Task List during run time; that is, either horizontal or vertical.

Controls	Value	Description
Font Size	User-defined	Provides for changing the size of the text font during run time.

Copy to Other Resources

Click to open the **Universal Browser** to select one or more computers, users, or personnel classes to copy the selected visibility options for the My Task List Editor to.

Preview

Displays the layout of the My Task List Editor based on the visibility selections. You can view the selections in either the Horizontal Layout Mode or the Vertical Layout Mode.

Global Subprocesses

Global Subprocesses

Global subprocesses allow you to develop and store standard and frequently used subprocesses. These subprocesses can be reused in various workflows. They are preconfigured and then are available in the Global Subprocesses list where you are creating your workflow.

When global subprocesses are used in a workflow, they cannot be changed from this location; they are read-only (as indicated by the lock symbol next to the subprocess). You can perform basic tasks, such as change the name and bind parameters, but the internal process cannot be edited. However, if you want to make a change to a global subprocess so that it is reflected in all workflows that use this subprocess, you can edit the original subprocess by selecting it from the Subprocesses resources folder (using the Universal Browser).

You can also choose to make a local copy of a global subprocess if you decide that the global subprocess does not fully meet the needs of the workflow. When you make a subprocess local, you can make changes that are reflected only in the current workflow. The Global Subprocesses list does not change.

In addition, if you have created a subprocess specific to a workflow and you want to re-use it in other workflows, you can make the subprocess global, which adds it to the Global Subprocesses list. However, you cannot make a local subprocess global if it has references to data outside the subprocess (excluding parameters). For example, if you are using a workflow level variable in a bind or expression, you must remove this external data reference before you are able to create a global subprocess. The subprocess must be self-contained for it to work as a global subprocess.

Global subprocesses are designed the same as local workflow subprocesses. They are configured in isolation and can contain any activity or variable that any manually created subprocess can have. These subprocesses can have input and output parameters that can be bound when they are added to a workflow, and they can also be composed of several user-defined activities. Global subprocess cannot be deleted if it is being used by a workflow or subprocess.

Add a subprocess from the Global Subprocesses list to a workflow

Before you begin

**Note:**

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. From **Global Subprocesses**, select the subprocess you want to add to the workflow.
7. Drag the subprocess to the appropriate drop point within the workflow.
 - a. To view the subprocess, double-click it.
 - b. To change the name of the subprocess, in the activity configuration panel, click **Edit**.

**Note:**

Changing the name of a global subprocess does not change the name of the original subprocess; it only changes the name within the current subprocess.

- c. To return to the workflow, click the **Workflow** link in the breadcrumbs.
8. Click **Save**.

Make a local subprocess global

If you are using a subprocess in a workflow that you would like to use again in another workflow, you can add it to the Global Subprocesses list.

Before you begin

**Note:**

A subprocess must be self-contained for it to work as a global subprocess; therefore, you must remove any reference to external data.

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. In the workflow, select the local subprocess you want to make global.
7. In the subprocess configuration panel, click **Make Global**.
The Make Subprocess Global dialog box appears.
8. Enter a name and, optionally, a description for the subprocess, and then click **Make Global**.

**Note:**

The new subprocess is added to the Global Subprocesses list and the Subprocesses resource list.

9. Click **Save**.

Make a local copy of a global subprocess

Before you begin

If you are using a global subprocess in a workflow and you want to make a change to it in order to better meet the needs of your workflow, you can make the subprocess local. This change does not affect the Global Subprocesses list. A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.

3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. In the workflow, select the global subprocess you want to make local.
6. In the subprocess configuration panel, click **Make Local**.

The **Make Subprocess Local** dialog box appears.

7. Click **Make Local**.
8. Click **Save**.

Index a directory resource against a global subprocess to be used during run time

Before you begin



Note:

If the workflow contains multiple global subprocesses, the index operation will return only the first instance that it finds.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. From the workflow configuration panel, click **Edit** next to Local Variables, and then click **Add**.
6. From the Type drop-down menu, select a variable type.



Tip:

- If you want to write to the local variable, from the Type field, select **Object**.
- If you want to read the value of the local variable, from the Type field, select a data type, and then in the Initial Value field, set a value.

7. Insert the Global Subprocess you want to work with.



Note:

You must have preconfigured global subprocesses available in your activities toolbox.

8. From the workflow, double-click a process step.
9. Drag an Index activity to the appropriate drop point.

10. On the activity configuration panel, select **Bind** from the drop-down arrow next to the Collection field.

The **Select Variable** dialog box appears.

11. From the Select Variable dialog box, under Workflow properties, select **Subprocesses**, and then click **OK**.

12. In the Index Type field, click the drop-down arrow and select **DirectoryResource**.

13. In the Index field, click the ellipsis button to select the Global Subprocess that you added in Step 5.

14. To read the results of the Index activity:

a. Drag a Write activity to a drop point after the Index activity.

b. Click the ellipsis button to select the variable that you want to read from the Index activity result, and then click **OK**.

c. In the Value field, from the drop-down arrow, select **Bind**.

d. From the Select Variable dialog box, select the local variable you added in Step 3 that you want to write to.

15. To write to the results of the Index activity:

a. Drag a Write activity to a drop point after the Index activity.



Note:

If you are using the same Write activity that you added in Step 12, then click **Add Write**.

b. Click the ellipsis button to select the local variable that you want to read from, and then click **OK**.

c. In the Value field, from the drop-down arrow, select **Bind**.

d. From the Select Variable dialog box, select a variable from the result of the Index activity.

16. Click **Save**.

Make Global/Local

When a global subprocess is inserted into a workflow, you can change it to a local subprocess. Then, you can configure the subprocess to meet the specific needs of the currently selected workflow.

There are properties that exist in the workflow or process scopes that are defined on a workflow's process step. They are supplied by the workflow engine and are available when authoring a global subprocess.

Basic Activities

Activities Used to Create Workflows

Activities comprise the main building blocks or steps that perform the actions in a subprocess or workflow.

Within each activity, there are a number of dependencies and relationships that can be created to other activities within subprocesses. The behavior of activities are configured through their properties and variables. Variables can also be used to output results from the activity's execution.

For example, some activities, such as the parallel activity, can have other activities placed inside them. By doing so, processes can be expanded and the branched activity can run more simultaneous activities to complete the required result.

Common Properties

You can configure the following common properties for each step contained within a workflow: name, description, and the enabled feature.

The steps in a workflow where you can configure these properties include: load, preprocess, presubprocess, subprocess, postsubprocess, postprocess, and unload. Setting these attributes allows you to identify an activity or a subprocess within a workflow, as well as decide whether or not to include the step in the workflow instance.

The activity and subprocess attributes consist of the following options:

Property	Value	Description
Name	User-defined	Enter a logical name that you want to assign to the subprocess or activity you are editing. Generally, the name of the subprocess relates to its purpose. For subprocesses, this name becomes the name of the step in the Task List.
Description	User-defined	Enter additional information related to a subprocess or activity. This may include instructions or

Property	Value	Description
		<p>detailed information on the purpose of these steps.</p> <div data-bbox="1024 380 1419 604" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;">  Note: The description is not displayed in the Task List. </div>
Enabled	System-defined	<p>Select this check box to include the activity in a workflow.</p> <div data-bbox="1024 737 1419 961" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;">  Note: You can only select this option within a sub-process or activity. </div>

Insert an activity

Before you begin



Note:

Activities can be inserted at both the workflow and subprocess levels; however, not all activities are available at both levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:

- At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

Edit activity and subprocess attributes

Before you begin

**Note:**

A resource must already have been added in order to select it from this list.

Prerequisites

[Add a resource to the workflow model \(on page 1395\)](#)

About this task

After you create a workflow, you can add and configure activities. Activities enable a workflow to process the tasks necessary to the successful completion of a workflow.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. On the configuration panel, next to the activity or subprocess name, click **Edit**.
The Edit Attributes dialog box appears.
6. In the **Name** field, enter a new name for the activity or subprocess.
7. Select the **Enabled** check box to be able to run the activity.
8. Optionally, in the **Description** field, enter additional information related to the activity or subprocess.
9. Click **OK**, and then click **Save**.

Delete an activity

Before you begin

**Note:**

When you delete a branching activity, such as If/Else or Parallel, all contained activities are also deleted.

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. In the workflow, right-click the activity you want to delete, and then select **Delete**.

Related information

[Add a resource to the workflow model \(on page 1395\)](#)

Move an activity

Before you begin

**Note:**

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. In the workflow, click the activity you want to move and drag it to an alternate drop point.

Related information

[Add a resource to the workflow model \(on page 1395\)](#)

Add a branch to an activity

Before you begin

Branching is used only with the following activities:

- If/Else
- Parallel

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with

**Note:**

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. Click the activity you want to branch.
7. In the workflow, right-click the activity, and then select **Add Branch**.
8. Click **OK**, and then click **Save**.

Related information

[Add a resource to the workflow model \(on page 1395\)](#)

Delete a branch from an activity

Before you begin

Branching is used only with the following activities:

- If/Else
- Parallel

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the branch within the activity that you want to remove.
6. On the **Workflow Editor** toolbar, click **Delete**.

**Note:**

This action only removes the selected branch. To remove the entire activity, see *Delete an activity*.

7. Click **Save**.

Related information

[Add a resource to the workflow model \(on page 1395\)](#)

[Delete an activity \(on page 1419\)](#)

Standard Activities

Activities define how each subprocess is executed. Task steps are created by placing multiple activities in a subprocess. The placement of these activities defines how the subprocess functions when a workflow runs.



Tip:

When you create subprocesses through the use of activities, it is important to review your subprocess to ensure that it functions as intended. The preferred method is by testing your workflow before putting it into production.

Workflow Level Activities

The If/Else and Parallel activities are available at both the subprocess and workflow levels. The Local Subprocess activity is available only at the workflow level.

Custom Activities

You can create your own user-defined activities for use in a subprocess. For more information, see *Third Party Activities*.

Related reference

[Third-party Activities \(on page 1422\)](#)

Related information

[Predefined User Activities \(on page 1426\)](#)

[Activities Used to Create Workflows \(on page 1416\)](#)

Third-party Activities

Workflow can utilize third-party activities that are compatible with Microsoft Windows Workflow Foundation.

With these elements, extensibility is greatly enhanced, and a variety of other options are available to create a workflow.

**Important:**

We do not directly support these activities. At the time of this printing, third-party activities have not been tested with our software. Use of third party activities should be properly tested prior to being implemented in a production environment.

Third-Party Activity Configuration**Important:**

If you are using third-party activities in a multi-server environment, you must install all associated files on each server using these activities.

To use third-party activities, you must create a file named:

```
ActivityExtensions.dita
```

The following sample configuration files demonstrate the structure for the XML document you must use to define the server, location of the file, and the assemblies used. The `AssemblyUri`, `Category`, and any `DependencyAssemblyUris` must be edited to reflect your third-party activity requirements.

File Location

After the file has been created, place the `ActivityExtensions.dita` file in the following location on the server (where your application was installed):

```
<installdir>\Proficy Workflow\Program\Data\Projects\SOAPProject\AppServers
\Workflow\Workflow
```

**Important:**

The activity assembly and its dependent assemblies **must** go in the following path in order for the URI to work: `\Proficy Workflow\Program\Applications\Workflow`

Example 1

The following is a sample configuration file for a single activity assembly with a single dependency:

```
<ActivityExtensions xmlns="http://schemas.datacontract.org/2004/07/Proficy.Workflow.Services">
  <ActivityExtension>
    <AssemblyUri>net.tcp://localhost/Program/Applications/Workflow/CustomActivityAssembly.dll</AssemblyUri>
    <Category>Custom Activities</Category>
```

```

<DependencyAssemblyUri>

<DependencyAssemblyUri>net.tcp://
localhost/Program/Applications/Workflow/CustomActivityAssembly.dll</DependencyAssemblyUri>

</DependencyAssemblyUri>

</ActivityExtension>
</ActivityExtensions>

```

Example 2

The following is a sample configuration file to use with two activity assemblies, one with two dependencies, and on one with no dependencies:

```

<ActivityExtensions xmlns="http://schemas.datacontract.org/2004/07/Proficy.Workflow.Services">

  <ActivityExtension>

    <AssemblyUri>net.tcp://localhost/Program/Applications/Workflow/CustomActivityAssembly1.dll</AssemblyUri>

    <Category>Custom Activity 1</Category>

    <DependencyAssemblyUri>

      <DependencyAssemblyUri>net.tcp://
localhost/Program/Applications/Workflow/CustomActivityAssembly2.dll</DependencyAssemblyUri>

      <DependencyAssemblyUri>net.tcp://
localhost/Program/Applications/Workflow/CustomActivityAssembly3.dll</DependencyAssemblyUri>

    </DependencyAssemblyUri>

  </ActivityExtension>

  <ActivityExtension>

    <AssemblyUri>net.tcp://localhost/Program/Applications/Workflow/CustomActivityAssembly2.dll</AssemblyUri>

    <Category>Custom Activity 2</Category>

    <DependencyAssemblyUri/>

  </ActivityExtension>

</ActivityExtensions>

```

Third-Party Activity Tag Definitions

The following table describes the different sections of the `ActivityExtensions.dita` file:

Tag	Value	Description
ActivityExtension	Contains AssemblyUri, Category, DependencyUris	Contains the defining tags for the required assemblies.
AssemblyUri	URI to assemblies	Defines the path to the assemblies.
Category	User-defined name	Defines the category the assembly will appear under in the Workflow activities list.
DependencyAssemblyUris	Contains the URI	Contains a list of dependency assembly URI elements.
DependencyAssemblyUri	URI to dependent assemblies	Defines the path to the dependent assemblies.

Further Information

To view more information on how to create custom activities for Microsoft Windows Workflow Foundation, using your Internet browser, navigate to the Microsoft MSDN library and search for: *Developing Workflow Activities*.

Related reference

[Common Properties \(on page 1416\)](#)

[Third-party Activity Examples \(on page 1425\)](#)

Related information

[Activities Used to Create Workflows \(on page 1416\)](#)

Third-party Activity Examples

In this example, there are two activity assemblies that are added to the system with a category of "Custom Activities":

First Assembly

- Assembly: ActivityAssembly1.dll
- Dependencies: ActivityAssembly1Dependency1.dll, ActivityAssembly1Dependency2.dll

Second Assembly

- Assembly: ActivityAssembly2.dll
- Dependencies: (none)

All assemblies need to be copied to the following location:

```
\Program Files\Proficy\Proficy Workflow\Applications\Workflow
```

ActivityExtensions.dita

The ActivityExtensions.dita file contains the necessary information and path locations for the 3rd party assemblies to be used with Workflow. Continuing our example utilizing ActivityAsseblly1.dll and ActivityAssembly2.dll from above, including their dependencies, the ActivityExtensions.dita file should contain the following code:

```
<ActivityExtensions xmlns="http://schemas.datacontract.org/2004/07/Proficy.Workflow.Services">
  <ActivityExtension>
    <AssemblyUri>net.tcp://localhost/Workflow/ActivityAssembly1.dll</AssemblyUri>
    <Category>Custom Activities</Category>
    <DependencyAssemblyUris>
      <DependencyAssemblyUri>net.tcp://localhost/Workflow/ActivityAssembly1Dependency1.dll</DependencyAssemblyUri>
      <DependencyAssemblyUri>net.tcp://localhost/Workflow/ActivityAssembly1Dependency2.dll</DependencyAssemblyUri>
    </DependencyAssemblyUris>
  </ActivityExtension>
  <ActivityExtension>
    <AssemblyUri>net.tcp://localhost/Workflow/ActivityAssembly2.dll</AssemblyUri>
    <Category>Custom Activities</Category>
    <DependencyAssemblyUris/>
  </ActivityExtension>
</ActivityExtensions>
```

Result

After the assembly DLL and DLL dependencies are placed, and the ActivityExtensions.dita is created, you must restart the Proficy Server service and Workflow client. Upon opening up a new workflow in the **Workflow Editor**, you will see your new assemblies below the Activity Toolbox.

Predefined User Activities

A user activity defines part of a user-defined process in a workflow process step. For a workflow to run, all user activities must be valid (contain no errors) and enabled (active).

User activities allow you to develop and store standard and frequently used activities. They are configured in isolation, can be reused in various subprocesses, and have the capability of taking input and output parameters. When a user activity is used by a workflow or subprocess, it cannot be deleted. It also cannot be edited if a workflow that uses the activity is running. A subprocess may be composed of several user activities.

When user activities are used in a subprocess, they cannot be changed; they are read-only (as indicated by the lock symbol next to the activity). You can perform basic tasks, such as change the local name and bind parameters, but the internal process cannot be edited. If you want to make a change to a user activity so that it is reflected in all subprocesses that use this activity, you can edit the original activity by selecting it from the User Activities resources folder in the Navigator.

You can also choose to make a local copy of a user activity if you decide that the global version does not fully meet the needs of the subprocess. When you make an activity local, you can make changes that are reflected only in the current copy of the activity. The User Activities list does not change.

In addition, if you have created a user activity specific to a subprocess and you want to re-use it in other workflows, you can make the activity global, which adds it to the User Activities list. However, you cannot make a local user activity global if it has references to data outside the user activity (excluding data passed into parameters). The user activity must be self-contained for it to work as a global activity.

User activities are displayed in the toolbox of the workflow editor after they are created in the User Activities resources folder. If you add a *category* to the user activity, it is displayed under this category; otherwise, it will be listed as Uncategorized. A user activity can have multiple categories assigned to it.

**Note:**

Only users with administrative access or with the appropriate *key set permissions* can add and configure user activities.

Add a user activity to a subprocess

When user activities are created in the resource list, they are available to be added to a subprocess or process step and configured.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.

5. Double-click the subprocess you want to add the user activity to.
6. Drag the user activity to the appropriate drop point within the subprocess.
 - a. To view the user activity, double-click it.
 - b. To change the name of the user activity, in the activity configuration panel, click **Edit**.

**Note:**

Changing the name of a global user activity does not change the name of the original user activity; it only changes the name within the current subprocess.

- c. To return to the subprocess, click the **Subprocess** link in the breadcrumbs.
7. If required, create an *expression* or perform a *bind* to set parameter values.
8. Click **Save**.

Related reference

[Workflow Binds \(on page 1286\)](#)

[Workflow Expressions \(on page 1286\)](#)

Related information

[Add a resource to the workflow model \(on page 1395\)](#)

Make a local copy of a global user activity

Before you begin

If you are using a global user activity in a subprocess and you want to make a change to it in order to better meet the needs of your workflow, you can make the user activity local. This change does not affect the user activity if it is used in other workflows. A resource must already have been added in order to select it from this list.

Procedure

1. In the Navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Double-click the subprocess you want to work with.
6. In the subprocess, select the user activity you want to make local.

7. In the configuration panel, click **Make Local**.
The Make User Activity Local message box appears.
8. Click **OK**.
9. Click **Save**.

Make a locally created user activity global

If you are using a user activity in a subprocess that you want to use again in another workflow, you can add it to the global User Activities resources list.

Before you begin

**Note:**

A user activity must be self-contained for it to work as a global activity; therefore, you must remove any reference to external data.

A resource must already have been added in order to select it from this list. When user activities are created in the resource list, they are available to be added to a subprocess or process step and configured.

Procedure

1. In the Navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. Double-click the subprocess you want to add the user activity to.
7. Drag the user activity to the appropriate drop point within the subprocess.
8. Configure the user activity.
9. In the configuration panel, click **Make Global**.
The Make Global dialog box appears.
10. Make the necessary changes, and then click OK.
11. Click **Save**.

Related reference

[User Activities Configuration Panel \(on page 1432\)](#)

Properties for User Activities

You can use properties to further define the resource you are configuring.

Properties

You can use the following properties to work with user activities in a workflow:

Property	Type	Description
Global User Activity	DirectoryResource	Specifies the global resource that defines the activity. This activity is null if the user activity is local. The property is read only.
Is Global User Activity	Boolean	Indicates that the activity is a global user activity. The property is read only.

Related reference

[Workflow Properties \(on page 1289\)](#)

Limits for User Activities

There are restrictions to the elements that comprise a workflow. These limits ensure a quality functioning workflow.

You can create a maximum of 500 user activities.

Limits

User activities are self-contained; therefore, these limits do not include the children of other activities. For example, if an existing user activity has 50 child activities, and it is used within a new user activity, it only counts as one child activity.



Important:

Data item limits include child activities.

The following table describes the available options.

Limit	Restriction	Result
Data items and External events (including contained activities)	100	Validation error.
Nesting	5 levels	Validation error.
Circular references	If X contains Y, then Y cannot contain X	Validation error.

Related reference

[Limits for Workflows \(on page 1292\)](#)

[Limits for Subprocesses \(on page 1377\)](#)

User Interface Attributes for User Activities

The **Overview** tab provides general information about the currently selected user activity.

Enable and disable a user activity

Before you begin

**Note:**

To edit a user activity, it must be disabled. A resource must already have been added in order to select it from this list.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. On the **Overview** tab:

Click...	To...
Enable	activate the user activity for execution.
Disable	deactivate the user activity for execution.

Related information

[Add a resource to the workflow model \(on page 1395\)](#)

User Activities Configuration Panel

In this configuration panel, you can further define user activity attributes.

Unlike other levels of the workflow, completion codes or fault handlers cannot be added directly to user activities. However, faults can be handled using the normal fault scope mechanism, or thrown to the calling activity.

Panel Attributes

**Tip:**

You can use a Write Activity to change some property values at run time.

Attribute	Value	Description
Parameters	User-defined	Click the Edit link to configure.
Local Variables	User-defined	Click the Edit link to configure.
Data Items	User-defined	Click the Edit link to configure.
External Events	User-defined	Click the Edit link to configure.
Namespaces	User-defined	Click the Edit link to configure.

Configure user activities

Procedure

1. In the navigator, click **Workflow**.
2. From the Folders list, select User Activities, and then select an activity.
3. Click the **Workflow Editor** tab.
4. To add an activity, drag it to the appropriate drop point within the user activity.

**Tip:**

For more information on configuring other workflow activities, see *Standard Activities*.

5. For attributes, click **Edit** next to the attributes collection you want to work with.

Related reference[Standard Activities \(on page 1422\)](#)

Data Access Activities

These activities are used for passing information and data between workflows and activities.

Call Method Activity

The Call Method activity allows you to use a call method from a service provider to access information in other areas of the application.

The outputs of the call method are available as variables to the rest of the workflow. For example, you can call a method in the production model to get the current status of a work request.

Input and Output Parameters

After selecting a call method from the **Configure Method** dialog box, configure the input and output parameters. You can *bind* workflow level variables to the inputs. Output parameters can be bound to local variables or output parameters of parent activities, and the values will be written to those variables when the method executes.

Enforce User Permissions

You can invoke a security check when service provider method calls are made within a workflow. When configuring a Call Method activity, go to the **Security** area to ensure that, at run time, the current user who has acquired the task step has permission to call the method.

**Note:**

For this feature to function correctly, you must configure a subprocess in one of the following ways to ensure that a user *acquires* the task step. Then, to verify that the task step is acquired, check the Current User property on the subprocess to ensure it is not null.

- *Manual Start* is set to `True`.
- A *Form activity* is used before the method call.

Related reference[Common Properties \(on page 1416\)](#)

Related information

[Activities Used to Create Workflows \(on page 1416\)](#)

[Set a task step to be automatically acquired \(on page 1390\)](#)

[Set a subprocess to Manual Start \(on page 1382\)](#)

[Configure a Form activity \(on page 1503\)](#)

Configure the Call Method activity

Prerequisites

[Add a resource to the workflow model \(on page 1395\)](#)

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, add a resource, or select the resource that you want to work with.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Drag and drop a **Local Subprocess** into the workflow and double-click it, or double-click an existing subprocess.



Note:

If required, configure the workflow.

6. Drag and drop a **Call Method** activity into the subprocess, or select an existing call method.
7. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
8. Double-click the service provider you want to use, and then select the call method.
9. Click **Next**.
10. In the **Item specific configuration** section, configure the method.



Note:

Item configuration depends on the selected source and item. Some items do not require configuration.

11. At run time, if you want to return all data from the selected item, select the **Return all data** option.
12. At run time, if you want to return a subset of data from the selected item at run time, select the **Return only a subset of data** option, and then from the data tree, select the data you want returned.

**Note:**

If the item returns only a single piece of data, the **Return all data** option is selected, and you cannot retrieve a subset of data.

13. Click **Finish**.
14. If required, on the activity configuration panel, modify the **Inputs/Outputs** and/or create an expression or perform a bind for the selected call method.
15. In the **Security** area, select the **Enforce User Permissions** check box.
16. Click **Save**.

Call Web Method Activity

The Call Web Method activity allows you to call an external web service method that contains parameters defined as part of the workflow.

**Important:**

The Call Web Method that was provided in earlier versions of Workflow is still available. You can continue to use it for your legacy web services; however, it is recommended that for all new web services configurations, you should use the Web Services Service Provider to add and configure web service definitions.

Two types of web services are supported:

- WCF (Windows Communication Foundation) web service, where the endpoint or interface name is clearly defined in the WSDL (Web Services Description Language) file.
- Conventional web services, where no endpoint or interface is defined in the WSDL file.

Web Method Details

The following table describes the web method details required to configure the Call Web Method activity:

Attribute	Value	Description
WSDL URL	User-defined	Specifies the URL of the WSDL file that contains the web service you want to call.

Attribute	Value	Description
		 Note: The URL must end with ? WSDL.
Interface/Endpoint	User-defined alphanumeric string (A-Z, a-z, 0-9)	Specifies the interface or endpoint the web service is monitoring.
Method Name	User-defined alphanumeric string (A-Z, a-z, 0-9)	Specifies the name of the web method being called.

Input and Output Parameters

After you configure the web method details, configure the input parameters. When you assign an input value, you can *bind* it to a variable to ensure that the parameter receives the data of the correct data type.

The output parameter is the return value of the call method; therefore, you do not need to enter a value.



Important:

Null values are not supported in a sequence of input parameters. For example, if three input parameters are required, the first and/or second parameters cannot be null. In this case, you can either use InputParam1 through InputParam3 for your input parameters, or you can use the *Expression Editor* to enter the numeral 0 or an empty string into the parameter field that you want to leave empty.

Code Activity

The Code activity allows you to write custom logic; therefore, you can augment the functionality of a workflow to meet your needs.

The Code activity only supports the language VB.NET. Using VB.NET, you can enter any valid snippet of code to accomplish specific needs based on the code entered. Using the Code activity, you can create parameters in the code body that can be bound to workflow fields, parameters, external variables, activity properties, and any additional bindable resources. However, you cannot directly access resources in the application, or any external resource.

**Important:**

When a code activity runs in a workflow, no other activity can run, even within a parallel activity. Therefore, avoid writing long-running code blocks.

To access external resources, running workflow resources must be passed in as parameters to the code. While a workflow can access only Workflow and main system assemblies, it can also import *namespaces* from external sources. All scripting within the Code activity is entered into the script editor.

**Important:**

When writing your code, ensure that you call `Dispose` on all objects that implement `IDisposable` created by your code. For example, if you open a file, you must also ensure that you close that file.

The following functions are supported for the Code activity:

Example supported Code functions	Unsupported Code functions
Variable declarations and assignments (Dim i as Integer; i = 5)	Declaring a class, interface, structure, or enum.
Math operations (i = 5*j -100)	Declaring a subprocedure or function.
Flow control logic (If/Else statements, While loops, goto's)	Declaring a namespace.
Calling member methods of a declared object, or an object passed in as a parameter (MyObj.DoSomething())	Importing a namespace.
Calling static methods on known/imported types	Returning a value: Code activities have no return values, they must provide output through code parameters (for example, you can call 'return' to indicate you want the snippet to end, but you cannot call 'return true').
Try-catch statements	Directly accessing properties of the workflow (activities, variables, and so on). You must pass any values you want to work with as parameters.
Throwing exceptions	Calling another code snippet in another activity.

Data Types

All system and application data types are supported. Importing namespaces is performed by using the namespace import functionality within a workflow definition or subprocess. The imported namespaces apply to all Code activities in the workflow or subprocess after they have been added. Further to this, Workflow allows you to add references from other assemblies.

Parameters

The Code activity parameters consist of subprocess and workflow parameters that have already been configured. To use elements of the workflow (for example, parameters, local variables, activity properties, and so on) in your code snippet, you must add a parameter and link it to the desired data. Changes to a parameter's value are reflected in the workflow after the code has been executed. For example, the parameters are 'ref' or 'InOut'.

Example Code Snippet

In this example, there are subprocess parameters called Field1 and Field2. In the Code activity configuration, Field1 will be bound to the parameter called InitialValue. Field2 will be bound to a parameter called AdjustedValue.

```
AdjustedValue = InitialValue * 2 + 40;  
  
If (AdjustedValue > 100) Then  
    AdjustedValue = 100  
  
End If
```

Upon running this script in an instance of a workflow, Field2 will be set to the value of AdjustedValue.

Related reference

[Common Properties \(on page 1416\)](#)

[Workflow Script Examples \(on page 1439\)](#)

[Namespaces \(on page 1354\)](#)

[Parameters \(on page 1343\)](#)

Related information

[Activities Used to Create Workflows \(on page 1416\)](#)

Workflow Script Examples

The following script examples are intended for example purposes only and do not necessarily perform any commonly used functions that users may use in their production environments.

Example 1

In this example, there is a subprocess local variable called Field1 and another called Field2. In the code activity configuration the parameter Field1 will be bound to the parameter called InitialValue. The Field2 field will be bound to a parameter called AdjustedValue. The following is an example of how this appears in the code activity editor:

```
AdjustedValue = InitialValue * 2 + 40;
If (AdjustedValue > 100) Then
    AdjustedValue = 100
End If
```

Upon running this script in an instance of a workflow, Field2 will be set to the value of AdjustedValue.

Example 2

In this example there are 2 parameters in the code activity: an Integer called InitialValue, which could be bound to a local variable or activity property, and a boolean called PerformFunction which is also bound to a local variable and serves as an output. The code snippet will perform a simple calculation on InitialValue and if the value is greater than 100, PerformFunction is set to true, which in turn, could drive a different execution path.

```
Dim adjustedValue As int32

adjustedValue = InitialValue * 2 + 40;
If (adjustedValue > 100) Then
    PerformFunction = True
Else
    PerformFunction = False
End If
```

Example 3

In this example, you can use a static string method to compare a variable to the string "Done". If they match then the code snippet will exit (using the Return statement). Otherwise, it will carry on and do something else indicated by the comment.

```

If String.Equals(MyString, "Done") Then
    Return
End If

' Comment: Do some more stuff here..

```

Add a code activity parameter

Before you begin



Note:

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Edit** next to Parameters.
The **Edit Code Parameters** dialog box appears.
7. Click **Add**.
8. In the **Name** and **Description** fields, enter a name and description for the activity.
9. From the **Select Variable** browser, select a value to bind to.

**Note:**

The resource properties available are dependent on the activities, parameters, data items, and local variables already configured. Not all options are available for each output value.

10. Click **OK**, and then click **Save**.

Related information

[Add a resource to the workflow model \(on page 1395\)](#)

Input code and insert a parameter

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Double-click the step containing the activity you want to work with.
6. Select the Code activity you want to insert a parameter in.

**Note:**

Parameters must already have been added to the Code activity.

7. Click the plus sign in the Code activity.
The script editor appears.
8. In the script editor, enter the code for this activity.
9. To insert a parameter within the code, click **Insert Parameter**. The **Insert Parameter** dialog box appears.
 - Select a parameter, and then click **OK**.
10. To validate the code, click **Validate**.

**Tip:**

If the code is invalid, the activity is shaded red, and a red exclamation mark appears, indicating which lines contain errors. If the Code activity is not red, then it is valid.

11. Click **Save**.

Related information

[Add a resource to the workflow model \(on page 1395\)](#)

Index Activity

An Index activity is used with any collection of data to look up a value at a specified index on the collection.

An Index activity allows you to select a single row or column out of a DataTable data type. Historian is a common source of data tables that can be accessed through *Data Items*. To access a cell of a data table, insert two Index activities. The first Index activity selects the row, and the second binds to the output of the first Index activity to select the column of that row.

Properties

The Index activity contains the following configuration options:

Property	Value	Description
Collection	System-defined	Use the drop-down arrow to <i>bind</i> a compatible resource property from an activity, the containing subprocess, a global subprocess, or a workflow. Click the ellipsis button to change these settings.
Index Type	System-defined	Allows you to choose the appropriate <i>data type</i> for the resource in the column or row to be selected. The available data types depend on the variable that you select in the Collection field.
Index	System-defined	Allows you to select a row, <i>bind</i> to another resource, or to create an <i>expression</i> based on the value.

Property	Value	Description
Outputs	Result; System-defined	Displays the output value of the Index activity. <div data-bbox="1024 390 1425 749" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;">  Note: If sufficient metadata is available, when indexing the rows collection of the DataTable, there is an output property for each column in the table. </div>

Related reference

[Common Properties \(on page 1416\)](#)

[Data Items \(on page 1348\)](#)

[Workflow Binds \(on page 1286\)](#)

[Workflow Expressions \(on page 1286\)](#)

Related information

[Activities Used to Create Workflows \(on page 1416\)](#)

Configure an Index activity

Before you begin**Note:**

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:

- At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, select **Bind** from the drop-down arrow next to the **Collection** field.
The **Select Variable** dialog box appears.
 7. From the **Select Variable** browser, select a value to bind to.



Note:

The resource properties available are dependent on the activities, parameters, data items, and local variables already configured. Not all options are available for each output value.

8. Click **OK**.
9. In the **Index Type** field, click the arrow and select a data type.
10. In the **Index** field, enter a literal value, or click the drop-down arrow to perform a bind or create an expression.
11. Click **Save**.

Index a directory resource against a global subprocess to be used during run time

Before you begin



Note:

If the workflow contains multiple global subprocesses, the index operation will return only the first instance that it finds.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. From the workflow configuration panel, click **Edit** next to Local Variables, and then click **Add**.
6. From the Type drop-down menu, select a variable type.



Tip:

- If you want to write to the local variable, from the Type field, select **Object**.
- If you want to read the value of the local variable, from the Type field, select a data type, and then in the Initial Value field, set a value.

7. Insert the Global Subprocess you want to work with.



Note:

You must have preconfigured global subprocesses available in your activities toolbox.

8. From the workflow, double-click a process step.
9. Drag an Index activity to the appropriate drop point.
10. On the activity configuration panel, select **Bind** from the drop-down arrow next to the Collection field.
The **Select Variable** dialog box appears.
11. From the Select Variable dialog box, under Workflow properties, select **Subprocesses**, and then click **OK**.
12. In the Index Type field, click the drop-down arrow and select **DirectoryResource**.
13. In the Index field, click the ellipsis button to select the Global Subprocess that you added in Step 5.
14. To read the results of the Index activity:
 - a. Drag a Write activity to a drop point after the Index activity.
 - b. Click the ellipsis button to select the variable that you want to read from the Index activity result, and then click **OK**.
 - c. In the Value field, from the drop-down arrow, select **Bind**.
 - d. From the Select Variable dialog box, select the local variable you added in Step 3 that you want to write to.
15. To write to the results of the Index activity:

- a. Drag a Write activity to a drop point after the Index activity.



Note:

If you are using the same Write activity that you added in Step 12, then click **Add Write**.

- b. Click the ellipsis button to select the local variable that you want to read from, and then click **OK**.
- c. In the Value field, from the drop-down arrow, select **Bind**.
- d. From the Select Variable dialog box, select a variable from the result of the Index activity.

16. Click **Save**.

Read Data Items Activity

The Read Data Items activity allows a subprocess to access data items outside of the workflow. This activity reads data from the data items defined at the workflow or subprocess level to make them available to other activities.



Note:

Data items must be configured with an Equipment model resource at either the workflow or subprocess level.

For more information, see *Data Items* and *Equipment model*.

Reads

Reads can be added to or removed from the Read Data Items activity. When you add a data item to the Read Data Items activity, a data item block is presented, displaying the input and output parameters. Input and output parameters vary depending on the data item you select. The parameters can be anything related to the data type of the data item selected.

Using the read value

To use the value from a Read Data Item activity in a workflow, you can create an *expression*, or you can *bind* an activity in the workflow to an output parameter of the Read Data Item after the Read Data Item activity in the workflow.

Configure a read for a Read Data Item activity

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Double-click the step containing the activity you want to work with.
6. Click the ellipsis button.
The **Select Data Item** dialog box appears.
7. Select the data item you want to configure.
The **Select Data Item** dialog box appears.
8. Click **OK**.
9. On the activity configuration panel, select **Bind** from the drop-down arrow next to the **Value** field.
The **Select Variable** dialog box appears.
10. Click the **Subprocess** or **Workflow** tab, and then select an output value to bind to.
11. Click **OK**, and then click **Save**.

Related information

[Add a resource to the workflow model \(on page 1395\)](#)

Write Activity

The Write activity performs a set of writes to workflow variables. The source of each write can be an expression, a bind, or a user-defined absolute value.

Write Activity Configuration

The variable browser for the Write activity displays all variables that are applicable in a given context. For example, in a subprocess you have access to all of the workflow variables (parameters, data items, and so on), all of the subprocess variables, and all of the activity properties. To add another layer, if you are at the fault handler level, then the properties of the fault handler are also available.



Note:

Data types must match when writing to parameters and variables, or when using expressions.

Configure a Write activity

Prerequisites

[Add a resource to the workflow model \(on page 1395\)](#)

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Double-click the step containing the activity you want to work with.
6. Click the ellipsis button.
The **Select Variable** dialog box appears.
7. Select the variable that the value is written to, and then click **OK**.



Note:

You must match the data type with the value to be written.

8. On the activity configuration panel, in the **Value** field, enter a value.
The **Select Variable** dialog box appears.
9. If required, create an *expression* or perform a *bind* to set parameter values.
10. Click **OK**, and then click **Save**.

Related reference

[Workflow Binds \(on page 1286\)](#)

[Workflow Expressions \(on page 1286\)](#)

Write Data Items Activity

The Write Data Items activity allows a subprocess or step to access data items outside of the workflow. The Write Data Items activity writes data to data items at the workflow or subprocess level.

Writes

Writes can be added to or removed from the Write Data Items activity. When adding a data item to the Write Data Items activity a data item block is presented, displaying the input and output parameters. By default, the input and output parameters are blank as configured by initial binding.

Input and output parameters vary depending on the data item you select. The parameters can be anything related to the data type of the selected data item.

Related reference

[Common Properties \(on page 1416\)](#)

[Data Items \(on page 1348\)](#)

[Workflow Literals \(on page 1289\)](#)

[Workflow Expressions \(on page 1286\)](#)

[Workflow Binds \(on page 1286\)](#)

Related information

[Activities Used to Create Workflows \(on page 1416\)](#)

Add a write to the Write Data Item activity

Before you begin

**Tip:**

More than one data item can be added to this activity.

**Note:**

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the Navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. Double-click the step containing the activity you want to work with.
7. On the activity configuration panel, click **Add Write**.
The **Select Data Item** dialog box appears.
8. Select the data item you want to configure.
The **Select Data Item** dialog box appears.
9. Click **OK**, and then click **Save**.

Related information

[Add a resource to the workflow model \(on page 1395\)](#)

Configure a write for a Write Data Item activity

Before you begin



Note:

A resource must already have been added in order to select it from this list. For more information, see [Add a resource to the workflow model](#).

Procedure

1. In the Navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. Double-click the step containing the activity you want to work with.
7. Click the ellipsis button.
The **Select Data Item** dialog box appears.
8. Select the data item you want to configure.
The **Select Data Item** dialog box appears.
9. Click **OK**.
10. On the activity configuration panel, in the **Value** field, enter a value.
11. If required, create an *expression* or perform a `to` to set parameter values.
12. Click **OK**, and then click **Save**.

Related reference

[Workflow Binds \(on page 1286\)](#)

[Workflow Expressions \(on page 1286\)](#)

Related information

[Add a resource to the workflow model \(on page 1395\)](#)

Remove a write from a Write Data Item activity

Before you begin

**Note:**

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the Navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. Double-click the step containing the activity you want to work with.
7. On the activity configuration panel, select the write you want to remove, and then click **Delete**.
8. Click **Save**.

Related information

[Add a write to the Write Data Item activity \(on page 1449\)](#)

Workflow Control Activities

These activities are used for creating changes in a workflow, such as starting or stopping activities, during run time.

Start Schedule Activity

The Start Schedule activity allows a workflow schedule to start during the execution of another workflow.

When you place a Start Schedule activity in a subprocess, step, fault scope, or fault handler, the activity runs when the activity is reached in the process. The Start Schedule activity can have parameters passed to and from it. If output parameters are being used, the activity can be configured to pause the workflow until the schedule that is called has been completed and the value for the output is available.

**Note:**

A workflow will terminate if the workflow that it is calling terminates, regardless of any related preconfigured settings, such as the Wait for the workflow to complete setting.

Settings Tab

You can use the Start Schedule activity to start a preconfigured workflow schedule from the workflow you are currently configuring.

Attribute	Value	Description
Schedule	User-defined	<p>Click the ellipsis button to select the schedule that you want to start.</p> <p>Click the arrow to:</p> <ul style="list-style-type: none"> • bind the schedule to a directory resource variable, or • reset the default value. <p>When you bind, you must select a template schedule.</p> <div data-bbox="1024 1241 1421 1509" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: If you do not select a template, your workflow will result in a validation error at runtime. </div>
Template Schedule	User-defined	<p>When you perform a schedule bind, you must select a schedule template that defines the input and output parameters.</p> <div data-bbox="1024 1734 1421 1885" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: The schedule and template schedule must </div>

Attribute	Value	Description
		<div data-bbox="1024 268 1419 464" style="border: 1px solid #0070C0; border-radius: 10px; padding: 5px;">  have matching input parameters; otherwise, at runtime, the workflow will fail. </div> <p data-bbox="1024 499 1419 575">Click the ellipsis button to select a template schedule.</p> <p data-bbox="1024 615 1419 690">Click the arrow to reset the default value.</p>
Wait for the workflow to complete	User-defined	<p data-bbox="1024 722 1419 890">Select this check box to allow a user to pause a workflow to wait for the workflow that it is calling to complete.</p> <p data-bbox="1024 930 1419 1098">When you select this check box, output parameters from the started schedule are available to bind to the calling workflow.</p> <div data-bbox="1024 1129 1419 1822" style="border: 1px solid #0070C0; border-radius: 10px; padding: 5px;"> <p data-bbox="1040 1140 1175 1192"> Note:</p> <ul data-bbox="1166 1241 1409 1766" style="list-style-type: none"> <li data-bbox="1166 1241 1409 1587">• If a workflow is calling a disabled or terminated workflow, it will also terminate regardless of the status of this property. <li data-bbox="1166 1602 1409 1766">• A workflow cannot be recursive; that is, it cannot call itself. </div>

Attribute	Value	Description
End the workflow if cancelled	User-defined; Literal	Select this check box to allow a user to end a schedule if the runtime activity is cancelled. Ending the schedule means that it stops running and, if <i>persistence</i> is enabled, terminates it. Cancelling the activity also ends the workflow and any child tasks. This action removes workflow results from the server to save space.

Assignments Tab

When you use this activity to start another workflow, you can override the preconfigured location and personnel assignments by selecting new location and/or personnel resources. You can also bind the assignment to a workflow property. The assignments displayed are related to the selected workflow schedule, as well as the workflow attached to that schedule.

Override

Assignments are configured in a workflow where you bind personnel and location properties. However, when you use this activity to start a schedule, you can override these assignments for the schedule you are starting. Depending on the schedule, you may be able to override assignments at the workflow level or the subprocess level.

The following table describes the available attributes:

Attribute	Value	Description
Override check box	User-defined	Select this check box to allow a user to override the location and/or personnel resources assigned to the schedule you want to start.
Location	User-defined	Click the ellipsis button to either: <ul style="list-style-type: none"> • edit the bind, or • select an alternate location to route the schedule.

Attribute	Value	Description
		<p>Click the arrow to change how the value is set:</p> <ul style="list-style-type: none"> • bind to a variable in the workflow, or • choose a literal location resource, or • reset the default value.
Personnel	User-defined	<p>Click the ellipsis button to either:</p> <ul style="list-style-type: none"> • edit the bind, or • select other personnel to route the schedule. <p>Click the arrow to change how the value is set:</p> <ul style="list-style-type: none"> • bind to a variable in the workflow, or • choose a literal personnel resource, or • reset the default value.

Related information

[Schedules \(on page 1530\)](#)

[Configure workflow or schedule instance behavior \(on page 1309\)](#)

Configure the Start Schedule activity

Prerequisites

[Add a resource to the workflow model \(on page 1395\)](#)

About this task**Note:**

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the **Settings** tab, in the **Schedule** field:
 - click the ellipsis button to select a schedule, or
 - from the drop-down arrow, select **Bind** to bind the schedule to a directory resource variable.



Note:

If you select **Bind**, you must select a template schedule.

7. In the **Template Schedule** field, click the ellipsis button to select a template schedule.
8. Configure the template's available input parameters to match the schedule's parameters.
9. If applicable, to clear field contents, click the drop-down arrow, and then select **Reset to Default**.
10. **Optional:** Select **Wait for the workflow to complete**, and then configure any available output parameters.
11. If applicable, select **Wait for the workflow to complete**, and then select **End the workflow if cancelled**.
12. Click **Save**.

Override location and personnel assignments in the Start Schedule activity

Before you begin



Note:

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the Settings tab, in the Schedule field:
 - click the ellipsis button to select a schedule, or
 - from the drop-down arrow, select **Bind** to bind the schedule to a directory resource variable.



Note:

If you select Bind, you must select a template schedule.

7. In the Template Schedule field, click the ellipsis button to select a template schedule.
8. Configure the template's available input parameters to match the schedule's parameters.
9. If applicable, to clear field contents, click the drop-down arrow, and then select **Reset to Default**.
10. Select **Wait for the workflow to complete**, and then configure any available output parameters.
11. Click **Save**.
12. Click the **Assignments** tab.

- a. To configure, select the **Override** check box associated with the workflow or subprocess.
 - b. To update the resource or to edit an existing bind, click the ellipsis button next to the **Location** or **Personnel** field.
 - c. To change the assignment value between *bind* or *literal*, click the drop-down arrow.
13. Click **Save**.

Related reference

[Workflow Binds \(on page 1286\)](#)

[Workflow Literals \(on page 1289\)](#)

Modify the input parameters of a Start Schedule activity

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Double-click the step containing the activity you want to work with.
6. To reconfigure input parameters:
 - a. In a **Parameter** field, enter a value, click the ellipsis button or link, or use the arrow to change your values.
 - b. From the drop-down list, perform a *bind* or create an *expression*.
 - c. To remove the value, click the arrow and select **Reset to Default**.
7. Click **Save**.

Related reference

[Workflow Binds \(on page 1286\)](#)

[Workflow Expressions \(on page 1286\)](#)

Start Workflow Activity

The Start Workflow activity allows one workflow to start another workflow at runtime.

Settings Tab

You can use the Start Workflow activity to start a preconfigured workflow from the workflow you are currently configuring.

**Note:**

A workflow will terminate if the workflow that it is calling terminates, regardless of any related preconfigured settings, such as the Wait for the workflow to complete setting.

Attribute	Value	Description
Workflow	User-defined	<p>Click the ellipsis button to select the workflow that you want to start.</p> <p>Click the arrow to:</p> <ul style="list-style-type: none"> • bind the workflow to a directory resource variable, or • reset the default value. <p>When you bind, you must select a template workflow.</p> <div data-bbox="1024 1052 1421 1316" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: If you do not select a template, your workflow will result in a validation error at runtime. </div>
Template Workflow	User-defined	<p>When you perform a workflow bind, you must select a workflow template that defines the input and output parameters.</p> <div data-bbox="1024 1545 1421 1898" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: The workflow and template workflow must have matching input parameters; otherwise, at runtime, the workflow will fail. </div>

Attribute	Value	Description
		<p>Click the ellipsis button to select a template workflow.</p> <p>Click the arrow to reset the default value.</p>
Wait for the workflow to complete	User-defined	<p>Select this check box to allow a user to pause a workflow to wait for the other that it is calling to complete.</p> <p>When you select this check box, output parameters from the current workflow are available to bind to the calling workflow.</p> <div data-bbox="1024 898 1419 1598" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note:</p> <ul style="list-style-type: none"> • If a workflow is calling a disabled or terminated workflow, it will also terminate regardless of the status of this property. • A workflow cannot be recursive; that is, it cannot call itself. </div>
End the workflow if cancelled	User-defined	<p>Select this check box to allow a user to end a workflow if the runtime activity is cancelled. Ending the workflow means that it stops running and, if <i>persistence</i> is enabled, terminates it. Cancelling</p>

Attribute	Value	Description
		the activity first ends any child tasks before the workflow itself ends. This action removes workflow results from the server to save space.

Assignments Tab

When you use this activity to start another workflow, you can override the preconfigured location and personnel assignments by selecting new location and/or personnel resources. You can also bind the assignment to a property of the workflow. The assignments displayed are related to the selected workflow.

Override

Assignments are configured in a workflow where you bind personnel and location properties. However, when you use this activity to start a workflow, you can override these assignments for the workflow you are starting. Depending on the workflow, you may be able to override assignments at the workflow level or the subprocess level.

The following table describes the available attributes:

Attribute	Value	Description
Override	User-defined	Select this check box to allow a user to override the location and/or personnel resources assigned to the workflow you want to start.
Location	User-defined	Click the ellipsis button to either: <ul style="list-style-type: none"> • edit the bind, or • select an alternate location to route the workflow or subprocess.
Personnel	User-defined	Click the ellipsis button to either:

Attribute	Value	Description
		<ul style="list-style-type: none"> • edit the bind, or • select other personnel to route the workflow or subprocess.

Example

You can use a Start Workflow activity to create production work requests. These production orders trigger a dispatching workflow, which gets the work request, reads the work process segments, and then executes the workflows that have been assigned to the process segment.

Related reference

[Common Properties \(on page 1416\)](#)

[Workflow Binds \(on page 1286\)](#)

[Workflow Literals \(on page 1289\)](#)

Related information

[Activities Used to Create Workflows \(on page 1416\)](#)

[Configure workflow or schedule instance behavior \(on page 1309\)](#)

Configure the Start Workflow activity

Prerequisites

[Add a resource to the workflow model \(on page 1395\)](#)

About this task



Note:

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:

- At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the **Settings** tab, in the **Workflow** field:
- click the ellipsis button to select a workflow, or
 - from the drop-down arrow, select **Bind** to bind the workflow to a directory resource variable.

**Note:**

If you select **Bind**, you must select a template workflow.

7. **Optional:** In the **Template Workflow** field, click the ellipsis button to select a template workflow.
8. Configure the template's available input parameters to match the workflow's parameters.
9. If applicable, to clear field contents, click the drop-down arrow, and then select **Reset to Default**.
10. Select **Wait for the workflow to complete**, and then configure any available output parameters.
11. If applicable, select **Wait for the workflow to complete**, and then select **End the workflow if cancelled**.
12. Click **Save**.

Override location and personnel assignments in the Start Workflow activity

Prerequisites

[Add a resource to the workflow model \(on page 1395\)](#)

About this task

**Important:**

The Assignments tab is not available when a bind is set on the workflow.

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the **Settings** tab, in the **Workflow** field:
 - click the ellipsis button to select a workflow, or
 - from the drop-down arrow, select **Bind** to bind the workflow to a directory resource variable.



Note:

If you select **Bind**, you must select a template workflow.

7. In the **Template Workflow** field, click the ellipsis button to select a template workflow.
8. Configure the template's available input parameters to match the workflow's parameters.
9. If applicable, to clear field contents, click the drop-down arrow, and then select **Reset to Default**.
10. Select **Wait for the workflow to complete**, and then configure any available output parameters.
11. Click **Save**.
12. Click the **Assignments** tab.
 - a. To configure, select the **Override** check box associated with the workflow or subprocess.
 - b. To update the resource or to edit an existing bind, click the ellipsis button next to the **Location** or **Personnel** field.
 - c. To change the assignment value between *bind* or *literal*, click the drop-down arrow.
13. Click **Save**.

Related reference

[Workflow Binds \(on page 1286\)](#)

[Workflow Literals \(on page 1289\)](#)

Modify input and output parameters of a Start Workflow activity

Prerequisites

[Add a resource to the workflow model \(on page 1395\)](#)

About this task**Note:**

You cannot modify a workflow's output parameters, but you can bind the parameters.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Double-click the step containing the activity you want to work with.
6. To reconfigure input parameters:
 - a. In a **Parameter** field, enter a value, click the ellipsis button or link, or use the arrow to change your values.
 - b. From the drop-down list, perform a *bind* or create an *expression*.
 - c. To remove the value, click the arrow and select **Reset to Default**.
7. To bind output values, click the arrow next to the **Parameter** field, and select **Bind**.

**Tip:**

Select **No Bind** to restore the default output.

8. Click **Save**.

Related reference

[Workflow Binds \(on page 1286\)](#)

[Workflow Expressions \(on page 1286\)](#)

End Task Activity

When designing a workflow, you can use this activity to cancel or complete a task that is currently executing in a running workflow.



Note:

Continuously looping workflows are not affected by this activity.

Attribute	Value	Description
Complete Task	User-defined; Boolean	Select this option to ensure that the workflow cancels all executing activities and transitions to completion. The postsubprocess and postprocess activities finish as expected before the task is completed.
Cancel Task	User-defined; Boolean	Select this option to ensure that the workflow cancels all executing activities and transitions to a cancelled state. When you select this option, the Cancel Reason appears.
Cancel Reason	User-defined; String	Associated with Cancel Task. Specifies a reason that the task was cancelled in order to provide more data on the operation. This value can be configured using a <i>literal value, expression, or bind</i> . If no information is entered into this field, the default reason is <code>Cancelled by system</code> .

Related reference

[Workflow Expressions \(on page 1286\)](#)

[Workflow Literals \(on page 1289\)](#)

[Workflow Binds \(on page 1286\)](#)

Configure an activity to end a task during run time

When designing a workflow, you can use this activity to cancel or complete a task that is currently executing in a running workflow.

Prerequisites

[Insert an activity \(on page 1417\)](#)

About this task**Tip:**

You can use other workflow activities, such as *Status Message Activity*, to prevent confusion for operators when a task ends unexpectedly.

Procedure

1. From a subprocess, drag and drop an End Task activity.
2. To cancel and complete all executing tasks in a workflow, select the **Complete Task** option.
3. To cancel all executing tasks in a workflow, select the **Cancel Task** option.
The **Cancel Reason** field is enabled.
4. In the **Cancel Reason** field, provide a reason the task was cancelled. This value can be configured using a *literal* value, *expression*, or *bind*.

**Note:**

If no information is entered into this field, the default reason is `Cancelled by system`.

5. Click **Save**.

Related reference

[Status Message Activity \(on page 1497\)](#)

[Workflow Binds \(on page 1286\)](#)

[Workflow Expressions \(on page 1286\)](#)

[Workflow Literals \(on page 1289\)](#)

Related information

[Bind a resource property to a workflow or subprocess \(on page 1284\)](#)

[Add an expression to a workflow or subprocess \(on page 1285\)](#)

End Task Step Activity

When designing a workflow, you can use this activity to cancel or complete a task step that is currently executing in a running workflow.

Attribute	Value	Description
Complete Task Step	User-defined; Boolean	<p>Select this option to ensure that the subprocess cancels all executing activities and transitions to completion.</p> <div data-bbox="1024 892 1417 1203" style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note: If this option is selected, form activities are cancelled regardless of any configuration within the Form activity.</p> </div>
Cancel Task Step	User-defined; Boolean	<p>Select this option to ensure that the subprocess cancels all executing activities and transitions to a cancelled state. When you select this option, the Cancel Reason appears.</p>
Cancel Reason	User-defined; String	<p>Associated with Cancel Task Step. Specifies a reason that the task step was cancelled in order to provide more data on the operation. This value can be configured using a <i>literal</i> value, <i>expression</i>, or <i>bind</i>. If no information is</p>

Attribute	Value	Description
		<p>entered into this field, the default reason is</p> <p>Cancelled by system</p> <p>.</p>

Related reference

[Form Activity \(on page 1499\)](#)

[Workflow Literals \(on page 1289\)](#)

[Workflow Expressions \(on page 1286\)](#)

[Workflow Binds \(on page 1286\)](#)

Configure an activity to end a task during run time

When designing a workflow, you can use this activity to cancel or complete a task that is currently executing in a running workflow.

Prerequisites

[Insert an activity \(on page 1417\)](#)

About this task



Tip:

You can use other workflow activities, such as *Status Message Activity*, to prevent confusion for operators when a task ends unexpectedly.

Procedure

1. From a subprocess, drag and drop an End Task activity.
2. To cancel and complete all executing tasks in a workflow, select the **Complete Task** option.
3. To cancel all executing tasks in a workflow, select the **Cancel Task** option.
The **Cancel Reason** field is enabled.
4. In the **Cancel Reason** field, provide a reason the task was cancelled. This value can be configured using a *literal* value, *expression*, or *bind*.



Note:

If no information is entered into this field, the default reason is `Cancelled by system`.

5. Click **Save**.

Related reference

[Status Message Activity \(on page 1497\)](#)

[Workflow Binds \(on page 1286\)](#)

[Workflow Expressions \(on page 1286\)](#)

[Workflow Literals \(on page 1289\)](#)

Related information

[Bind a resource property to a workflow or subprocess \(on page 1284\)](#)

[Add an expression to a workflow or subprocess \(on page 1285\)](#)

Message Queue Activities

These activities are used for working with a messaging queue in order to co-ordinate work between workflows and activities.

Create Message Queue Activity

This activity creates a message queue for sending or receiving messages between workflows or activities in order to coordinate work.

This activity creates a `WorkflowQueue` using the `WorkflowQueuingService`.



Note:

The following restrictions apply:

- The message queue name cannot be null or empty, and you cannot use workflow instance ID, task instance ID, or task step instance ID as a queue name, because doing so causes the activities to fault at run time.
- Empty GUIDs are not allowed.



- Leading and trailing spaces are trimmed in the queue name; otherwise, the queue name is invalid.
- If the queue name is not a reserved value but the queue already exists, no fault occurs.

Property	Value	Description
Queue Name	User-defined; String	Specifies the name of the message queue, which is scoped to this workflow instance.

Configure a message queue

This activity creates a message queue for sending or receiving messages between workflows or activities in order to coordinate work.

Prerequisites

[Insert an activity \(on page 1417\)](#)

About this task

This activity creates a WorkflowQueue using the WorkflowQueuingService.

Procedure

1. From a subprocess, drag and drop a Create Message Queue activity.
2. In the **Queue Name** field, provide a name for the message queue. This value can be configured using a *literal* value, *expression*, or *bind*.



Note:

The following restrictions apply:

- The message queue name cannot be null or empty, and you cannot use workflow instance ID, task instance ID, or task step instance ID as a queue name, because doing so causes the activities to fault at run time.
- Empty GUIDs are not allowed.
- Leading and trailing spaces are trimmed in the queue name; otherwise, the queue name is invalid.
- If the queue name is not a reserved value but the queue already exists, no fault occurs.

3. Click **Save**.

Related reference[Workflow Binds \(on page 1286\)](#)[Workflow Expressions \(on page 1286\)](#)[Workflow Literals \(on page 1289\)](#)**Related information**[Bind a resource property to a workflow or subprocess \(on page 1284\)](#)[Add an expression to a workflow or subprocess \(on page 1285\)](#)

Send Message to Queue

This activity allows an author to configure a workflow to send a message to a queue by specifying the workflow instance ID, queue name, message type, and message in order to coordinate work between workflows and activities.

This activity interacts with a `WorkflowQueue` using the `WorkflowQueuingService`.

**Note:**

The following restrictions apply:

- The message queue name cannot be null or empty, and you cannot use workflow instance ID, task instance ID, or task step instance ID as a queue name, because doing so causes the activities to fault at run time.
- Empty GUIDs are not allowed.
- Leading and trailing spaces are trimmed in the queue name; otherwise, the queue name is invalid.
- A queue must be created before sending a message.
- The workflow instance should be running before sending a message. If it is not running, the instance is not loaded/restarted.

Queue Configuration Properties

You are required to configure these properties for this activity.

Property	Value	Description
Queue Name	User-defined; GUID; String	Specifies the name of the message queue, which is scoped to

Property	Value	Description
		this workflow instance. This value can be configured using a <i>literal value, expression, or bind</i> .
Workflow Instance ID	User-defined	Specifies the unique identifier that is scoped to the queue name. The default value of the property must be bound to the Workflow.Instance ID property.

Message Input Properties

Property	Value	Description
Message Type	User-defined	Specifies the data type of the message being sent, which is scoped to this workflow instance.
Message	Depends on Message Type	Specifies the message being sent to another queue, which can be bound to the Receive Message from Queue activity. This value can be configured using a <i>literal value, expression, or bind</i> .

Related reference

[Workflow Literals \(on page 1289\)](#)

[Workflow Expressions \(on page 1286\)](#)

[Workflow Binds \(on page 1286\)](#)

Configure a workflow to send a message to a queue during run time

Prerequisites

[Insert an activity \(on page 1417\)](#)

Procedure

1. From a subprocess, drag and drop a Create Message Queue activity.
2. In the **Queue Name** field, provide a name for the message queue. This value can be configured using a *literal* value, *expression*, or *bind*.
3. **Optional:** To change the default **Workflow.Instance ID** property, in the **Workflow Instance ID** field, click the ellipsis button to bind this property to another workflow GUID or property.
4. From the **Message Type** drop-down list, select the data type related to the message being sent.
5. In the **Message** field, enter the message you want to send to another workflow or activity.
6. Click **Save**.

Related reference

[Workflow Binds \(on page 1286\)](#)

[Workflow Expressions \(on page 1286\)](#)

[Workflow Literals \(on page 1289\)](#)

Related information

[Bind a resource property to a workflow or subprocess \(on page 1284\)](#)

[Add an expression to a workflow or subprocess \(on page 1285\)](#)

Receive Message from Queue

This activity allows an author to configure a message to be received from a queue, which was sent via the Send Message to Queue activity, in order to coordinate work between workflows and activities.

This activity interacts with a WorkflowQueue using the WorkflowQueuingService.



Note:

The following restrictions apply:

- The message queue name cannot be null or empty, and you cannot use workflow instance ID, task instance ID, or task step instance ID as a queue name, because doing so causes the activities to fault at run time.
- Empty GUIDs are not allowed.
- Leading and trailing spaces are trimmed in the queue name; otherwise, the queue name is invalid.



- A queue must be created before sending a message.
- The workflow instance should be running before sending a message. If it is not running, the instance is not loaded/restarted.

Queue Configuration Properties

Property	Value	Description
Queue Name	User-defined; GUID; String	Specifies the name of the message queue, which is scoped to this workflow instance. This value can be configured using a <i>literal value, expression, or bind</i> .
Time Out	User-defined	Specifies the duration that the message is in the queue before it is no longer valid. If this activity times out, it will be cancelled. As the default value is zero, if this property is not configured, this activity does not end; it will wait forever.

Message Input Properties

Property	Value	Description
Message Type	User-defined	Specifies the data type of the message received, which is scoped to the Send Message to Queue activity.

Message Output Properties

Property	Value	Description
Message	Read Only; Object	Specifies the message being received from another queue,

Property	Value	Description
		which is bound to the Send Message from Queue activity. This value can be configured using a <i>literal value, expression, or bind</i> .
Message Received	Read Only; Boolean	Specifies whether the sent message has been received.
Timed Out	Boolean	Specifies whether the received message is no longer valid because the allotted pre-set time to view the message has expired.

Related reference

[Workflow Literals \(on page 1289\)](#)

[Workflow Expressions \(on page 1286\)](#)

[Workflow Binds \(on page 1286\)](#)

Configure a workflow to receive a message from a queue during run time

This activity allows an author to configure a message to be received from a queue, which was sent via the Send Message to Queue activity, in order to coordinate work between workflows and activities.

Prerequisites

[Insert an activity \(on page 1417\)](#)

About this task

This activity interacts with a WorkflowQueue using the WorkflowQueuingService.

Procedure

1. From a subprocess, drag and drop a Receive Message from Queue activity.
2. In the **Queue Name** field, provide a name for the message queue. This value can be configured using a *literal value, expression, or bind*.
3. **Optional:** In the **Time Out** field, use the up and down arrows to set the duration that the message remains in the queue before it expires. You can set the number of days, hours, minutes, and/or seconds.
4. From the **Message Type** drop-down list, select the data type related to the message being received.

5. The following fields are informational, based on the interactions with the Send Message to Queue activity.
 - a. The **Message** field displays the message being received from another workflow or activity.
 - b. The **Message Received** field indicates whether the sent message has been received.
 - c. The **Timed Out** field indicates whether the sent message expires. It is *True* if the activity times out.
 6. Click **Save**.
-

Related reference

[Workflow Binds \(on page 1286\)](#)

[Workflow Expressions \(on page 1286\)](#)

[Workflow Literals \(on page 1289\)](#)

Related information

[Bind a resource property to a workflow or subprocess \(on page 1284\)](#)

[Add an expression to a workflow or subprocess \(on page 1285\)](#)

Delete Message Queue Activity

This activity deletes a message queue when it is no longer required.

This activity interacts with a `WorkflowQueue` using the `WorkflowQueuingService`.



Note:

The following restrictions apply:

- The message queue name cannot be null or empty, and you cannot use workflow instance ID, task instance ID, or task step instance ID as a queue name, because doing so causes the activities to fault at run time.
- Empty GUIDs are not allowed.
- Leading and trailing spaces are trimmed in the queue name; otherwise, the queue name is invalid.
- If the queue name is not a reserved value, but the queue already exists, no fault occurs.

Property	Value	Description
Queue Name	User-defined; GUID	Specifies the name of the message queue, which is scoped to this workflow instance.

Related reference

[Workflow Literals \(on page 1289\)](#)

[Workflow Expressions \(on page 1286\)](#)

[Workflow Binds \(on page 1286\)](#)

Remove a message queue from the system

You can delete a message queue and purge all remaining messages.

Prerequisites

[Insert an activity \(on page 1417\)](#)

Procedure

1. From a subprocess, drag and drop a Delete Message Queue activity.
2. In the **Queue Name** field, provide a name for the message queue. This value can be configured using a *literal* value, *expression*, or *bind*.
3. Click **Save**.

Related reference

[Workflow Literals \(on page 1289\)](#)

[Workflow Expressions \(on page 1286\)](#)

[Workflow Binds \(on page 1286\)](#)

Related information

[Bind a resource property to a workflow or subprocess \(on page 1284\)](#)

[Add an expression to a workflow or subprocess \(on page 1285\)](#)

Flow Control Activities

These activities are used to create multiple execution layers within workflow logic.

Delay Activity

The Delay activity allows a pause or a time out in a sequence or process while a workflow is running. When the activity has met the specified criteria, the process or sequence can then complete the next set of activities assigned to the process.

The Delay activity can be configured in days, hours, minutes, or seconds. The time-out value can also be bound to a subprocess or workflow variable, or used in an expression, which can be used to get TimeSpan values for the delay.

Configure a Delay activity

Procedure

1. Insert an activity.
2. On the activity configuration panel, in the **Timeout** field, use the arrows to set the delay.
3. If applicable, to perform a *bind* or reset, click the drop-down arrow.
4. Click **OK**, and then click **Save**.

Related reference

[Workflow Binds \(on page 1286\)](#)

Related information

[Insert an activity \(on page 1417\)](#)

Break Activity

The Break activity ends the execution of For Each and Parallel activities, exiting the activity at the point following the break. So, if there are any activities following the Break activity, they do not execute.



Note:

When using a Break activity, it must be contained within a For Each or Parallel activity; otherwise, the workflow becomes invalid.

Related reference

[Common Properties \(on page 1416\)](#)

While Activity

The While activity repeats the execution of a workflow subsection as long as a specified condition is true. The condition is checked each time the activity begins.

When configuring a While activity, ensure that the While condition eventually resolves to False; otherwise, the execution of the While branch will repeat indefinitely.

Properties of activities inside a While loop are not available to activities outside of the loop. If values in a While loop need to be accessed, they can be stored in local variables allowing *binding* from activities outside the local variable.



Important:

A condition must be set on the While loop in order to be valid.

Conditions

While activity conditions are created through the *Workflow Expressions* builder. Enter a valid expression for the condition using a combination of the functions, variables, completion codes, and operators.

Tab	Values	Description
Operators	User-defined	Enter a valid expression to calculate or evaluate the results to equal true.
Expressions	System-defined	A default set of functions that can be added to the expression.
Workflow Variables	User-defined	Parameter, activity, and workflow properties that can be added to the expression.
Completion Codes	User-defined	Select the preconfigured completion code suitable for the While loop.

Configure a While activity

Before you begin



Note:

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Edit Condition**.
The **Expression Editor** dialog box appears.
 7. Create the expression required to satisfy the While condition:
 - a. Click **Check Syntax**.
 - b. Click **Test**.
 - c. Click **Evaluate**.
 - d. Click **Close**, and then click **OK**.
 8. If required, click and drag an activity you want to insert onto the While activity drop point, and configure it.
 9. Click **Save**.

Results



Important:

As long as the While activity expression evaluates to True, it continues to loop. As soon as the condition evaluates to False, the process continues.

Related information

[Add a resource to the workflow model \(on page 1395\)](#)

Parallel Activity

A Parallel activity contains two or more sequences that execute at the same time. All branches complete before the activity completes, unless a Break activity is used.



Note:

If you are opening an existing Parallel activity at the subprocess level from a version of Workflow prior to 2.1 SP1, the Parallel activity is automatically updated to a newer version of this activity in order to support the new functionality contained in the Break activity.

Related reference

[Common Properties \(on page 1416\)](#)

[Visible Property \(on page 1382\)](#)

Add and modify a parallel activity

A Parallel activity contains two or more sequences that execute at the same time. All branches complete before the activity completes, unless a Break activity is used.

Prerequisites

[Add a resource to the workflow model \(on page 1395\)](#)

About this task



Note:

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. To add an activity to a branch, from the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
 7. To add another branch to a Parallel activity, select the activity, and then click .

A new branch is added to the right.
 8. **Optional:** If you have added this activity to a subprocess, add a Break activity to end its execution within this activity.
 9. Click **Save**.

For Each Activity

A For Each activity repeats the execution of a section of the workflow several times for a number of target instances in a collection.

Execution of a For Each branch activity works as follows:

1. At the beginning of the For Each activity, a For Each variable local to that branch is set to the first instance in the collection.
2. Execution continues to the end of the For Each activity. Activities refer to the For Each variable to perform operations with or on the current instance.
3. At the end of the For Each activity:

- If all instances in the collection are exhausted, execution continues after the For Each activity.
- Otherwise, execution returns to the beginning of the For Each activity and the local For Each variable is set to the next item in the collection.



Note:

The order in which instances are chosen from the collection cannot be configured.

General Settings

Inputs	Description	
Collection		<p>Displays IEnumerable. This is defined by the workflow and can be <i>bound</i> to parameters, local variables, properties of other activities including outputs of previously executed activities, and external variables having compatible types.</p> <div data-bbox="1024 1108 1412 1554" style="border: 1px solid #00a0c0; border-radius: 10px; padding: 10px;"> <p> Note: When binding to a Data Table returned by a data item read, if the For Each is bound to the rows collection of the DataTable, there is an output property for each column in the table.</p> </div>

Outputs	Description	
Current Item		Displays as object type. This is defined by the contained activities.

Example

For Each can repeat all subprocesses by selecting the workflow's subprocesses property as the Items Collection for iteration.

Related reference

[Workflow Binds \(on page 1286\)](#)

[Common Properties \(on page 1416\)](#)

[Parameters \(on page 1343\)](#)

[Local Variables \(on page 1345\)](#)

Configure a For Each activity

Before you begin

**Note:**

Activities can be inserted at both the workflow and subprocess levels.

Prerequisites

[Add a resource to the workflow model \(on page 1395\)](#)

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:

- Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, select **Bind** from the drop-down arrow next to the **Collection** field.
The **Select Variable** dialog box appears.
 7. From the **Select Variable** browser, select a value to bind to.

**Note:**

The resource properties available are dependent on the activities, parameters, data items, and local variables already configured. Not all options are available for each output value.

8. Click **OK**.
9. Click and drag an activity you want to insert onto the For Each activity drop point, and configure it.
10. **Optional:** If you have added this activity to a subprocess, add a Break activity to end its execution within this activity.
11. Click **Save**.

Related reference

[Break Activity \(on page 1479\)](#)

If/Else Activity

The If/Else activity changes execution of the workflow to one of multiple exit branches based on the resolution of a condition.

An If/Else branch can have multiple exit branches, each of which has a condition. Conditions on multiple exit flows are calculated from left to right. The flow executes on the first exit branch whose condition evaluates to True. There are no conditions on the last branch of the activity, as it is automatically set to True.

**Tip:**

The subprocess property value *Visible Property* can be used for this activity.

If/Else Branch

The If/Else branch contains one or more sequences as branches. Each branch has a condition that is evaluated to a boolean result. If the condition evaluates to True, the sequence for that branch is queued

as the next activity for the workflow instance to execute. Conditions are evaluated from left to right, and the first sequence with a condition evaluating to True is executed. If no condition is specified, the branch serves as an Else and always executes.

Add a condition to an If/Else activity

Before you begin



Note:

There can be only one condition placed on a branch of an If/Else activity, with the exception of the last branch.

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. Click the branch you want to create the condition on.
7. On the activity configuration panel, click **Edit Condition**.

The **Expression Editor** dialog box appears.
8. Enter the expression required to satisfy the If/Else condition:

**Tip:**

For more information, see *Workflow Expressions*.

- a. Click **Check Syntax**.
 - b. Click **Test**.
 - c. Click **Evaluate**.
 - d. Click **Close**, and then click **OK**.
9. Click **Save**.

Related reference

[Workflow Expressions \(on page 1286\)](#)

Related information

[Add a resource to the workflow model \(on page 1395\)](#)

Event-based Activities

These activities are used when you require workflow logic to start an event during run time.

Create Event Activity

The Create Event activity allows you to create a condition event for a workflow.

This event is triggered when the workflow is executing. Also, the output property of the event provides the subscription resource that is used by the *Wait for Event Activity*.

Configuring the Create Event activity is similar to creating a condition event in Workflow; however, there are some differences. For variables, when you bind expression variables to external data, you must choose a data item from the workflow. For expressions, workflow variables and completion codes are now accessible.

For more information on:

- creating conditional events, see *Condition Events*.
- building expressions, see the expression editor *Expression Editor*.
- building condition event expressions, see *Condition Expressions*.

Configure a Create Event activity

Before you begin



Note:

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Drag and drop a **Local Subprocess** into the workflow and double-click it, or double-click an existing subprocess.
6. On the activity configuration panel, click **Configure Event** next to **Conditional Event**.
The **Configure Conditional Event** dialog box appears.
7. Create a condition event.



Note:

For more information, see *Condition Events*.

8. Click **OK**, and then click **Save**.

Wait for Event Activity

The Wait For Event activity waits for one or more events to occur, based on the events added to this activity; otherwise, it can be set to time out.

The Wait For Event activity can subscribe to events created in the Event model. To use these events in the current workflow's Wait for Event activity, they must be added to *External Events* in the subprocess or workflow configuration panels.

The Wait For Event activity can also subscribe to local events created in a workflow by the *Create Event Activity*, or may trigger *Event Handlers*.

General Settings

The following table describes the wait behaviors available for this activity:

Wait for...	Value	Description
All events to occur	N/A	Select this option if you want the activity to wait for all events in the events list to occur before moving to the next activity.
Any event to occur	N/A	Select this option if it does not matter which event in the events list occurs before moving to the next activity.
Timeout	User-defined TimeSpan (days, hours:minutes:seconds)	Specifies a timeout condition where the activity waits for one or more activities to occur, or timeout at the specified time. You can <i>bind</i> to other workflow, subprocess, or activity properties, or build an <i>expression</i> based on existing workflow values.

Events

Attribute	Value	Description
Add Event	N/A	Click this button if you want to add more than one event to the Events list.
Ellipsis	User-defined	Click this button to select an event resource.
Remove Event button	N/A	Click this button to remove the current event resource.
Remove Event button	N/A	Click this button to remove the current event resource.
Event Source	System-defined	Displays the name of any pre-configured event that has been added.

Attribute	Value	Description
Outputs: Occurred	System-defined	The default is a Boolean data type that can be bound to a workflow, subprocess, or activity variable.
Outputs: other	User-defined	Output properties are dependent on the configuration of selected events.

Property Values

The following property values are available for this activity:

Property	Value	Description
TimedOut	Boolean	Indicates whether the event timed out or not. Output is True or False.
Timeout	TimeSpan	Indicates the time to time out.
WaitForAll	Boolean	Indicates if this activity will wait for all events to occur. Output is True or False.
WaitDefinition	System-defined	Output based on configuration of the wait definition.

Related reference

[Workflow Binds \(on page 1286\)](#)

[External Events \(on page 1353\)](#)

[Create Event Activity \(on page 1488\)](#)

[Event Handlers \(on page 1514\)](#)

[Workflow Expressions \(on page 1286\)](#)

Configure a Wait For Event activity

Before you begin



Note:

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. Double-click the step containing the activity you want to work with.
 7. In the **Wait For** area, select one of the following options:
 - All event to occur.
 - Any events to occur.
 8. If required, in the **Timeout** field, use the arrows to set the timeout.
The **Select Variable** dialog box appears.
 9. If applicable, to perform a *bind* or reset, click the drop-down arrow.
 10. Click **OK**.
 11. Click the ellipsis button.
The **Select Event** dialog box appears.
 12. Select an event, and then click **OK**.
 13. If applicable, to perform a *bind* or reset, click the drop-down arrow.
The **Select Variable** dialog box appears.

14. From the **Select Variable** browser, select a value to bind to.

**Note:**

The resource properties available are dependent on the activities, parameters, data items, and local variables already configured. Not all options are available for each output value.

15. Click **OK**, and then click **Save**.

Related reference

[Workflow Binds \(on page 1286\)](#)

Add an event to the Wait For Event activity

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. On the activity configuration panel, click **Add Event**.
The **Select Event** dialog box appears.
5. Select an event, and then click **OK**.
6. In the **Wait For** area, select one of the following options:
 - All events to occur.
 - Any events to occur.
7. If required, in the **Timeout** field, use the arrows to set the timeout.
The **Select Variable** dialog box appears.
8. From the **Select Variable** browser, select a value to bind to.

**Note:**

The resource properties available are dependent on the activities, parameters, data items, and local variables already configured. Not all options are available for each output value.

9. Click **OK**.
10. Select **No Bind** to restore the default output.
11. Click **Save**.

Keyword Activities

You can use keywords to replace text in work instructions and email subjects and bodies. This text is replaced with keyword values in a workflow at run time.

Replace keyword text during run time

You can use keywords to replace text in work instructions and email subjects and bodies. This text is replaced with keyword values in a workflow at run time.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Drag and drop a **Local Subprocess** into the workflow and double-click it, or double-click an existing subprocess.



Note:

If required, configure the workflow.

6. There are two types of keyword activities you can use.

Insert...	To...
Replace Keywords	replace keywords in a block of text. Go to step 7.
Update Work Instruction Keywords	ensure that the change in the value of a workflow variable is reflected in the work instruction text. Go to step 8.

7. To use the Replace Keywords activity, perform the following procedure.

- a. Insert a Replace Keywords activity.
- b. In the **Keyword Text** field, bind the activity to the property whose text you want to replacement .
The activity's output is the text that was defined to replace the keyword.

- To automatically update keywords in work instructions when the workflow begins, insert the Update Work Instruction Keywords activity.

**Note:**

This activity is used only if the values in a work instruction should be updated during the execution of a task step.

- Click **Save**.

Replace Keywords Activity

You can use keywords to replace text in email subjects and bodies. This text is replaced with keyword values in a workflow at run time.

Inputs

Inputs	Value	Description
Keyword Text	String	Specifies the text that you want to replace. Use the following format: <@Keyword@>.

Outputs

Outputs	Value	Description
Result Text	String	Displays the text that has been replaced with the keyword value.

Related reference

[Keyword Properties \(on page 1496\)](#)

Related information

[Replace keyword text during run time \(on page 1494\)](#)

Update Work Instruction Keywords Activity

This activity updates task step work instructions with current keyword values during run time.

You use this activity if the value of a work instruction requires updating during the execution of a task step. Then, the work instruction is automatically updated when the workflow begins and any task step begins.

Keyword Properties

Keywords are derived from local variables and properties configured in a workflow. Keyword property values are substituted during run time.

You can add keywords in two ways:

- Use the **Keywords** tab in the **Template Editor**
- Use a *Replace Keywords* or *Update Work Instruction Keywords* activity

You can add keywords using a *Replace Keywords* or *Update Work Instruction Keywords* activity in a workflow.

Additionally, custom keywords can be defined as *local variables* within a workflow or subprocess with the variable name referenced directly inside the standard keyword delimiters. Keywords are resolved at the lowest possible scope. For example, if a workflow and a subprocess both have a **Name** property, the `<@Name@>` keyword in a subprocess will be substituted with the subprocess name since that is the immediate scope, not the workflow name (the parent scope).



Note:

The list of keywords associated with a task template is a reference available for Task Builders. It does not make a keyword value available during run time unless the keyword name maps to an existing property or local variable.

Workflow Properties

For a list of properties that can be used with keywords, see *Workflow properties*.

Subprocess Properties

For a list of properties that can be used with keywords, see *Properties for subprocesses*.

Operator Activities

These activities are used by operators when a workflow is running the Task List.

Status Message Activity

The Status Message activity allows you to create custom status messages throughout the instance of a workflow.

The Status Message activity allows you to create custom status messages which are visible when a workflow is running. By using status messages, you provide information about a variety of considerations related to the task to the operators performing these tasks; for example, long processing times. Each subsequent status message activity overwrites the previous message.



Tip:

The property value of the containing subprocess must be *Visible Property* in order for this activity to be valid, and a message must be entered.

General Settings

Option	Value	Description
Update Status	User-defined	Updates the status message by overwriting the previous status message in Task List/Task Management.
Clear Status	User-defined	Clears the status message.

Example

Waiting for unit 'x' to finish processing.

X is an equipment name obtained from a Production Service Provider call method. For this message to display, you require a Call Method activity and Index activity to pull the correct data into the location where the Status Message activity is.

Perform the following actions:

1. Use a *Call Method Activity* to call the "ReadEquipmentRequirements" method on the Production service provider for a specified segment ID. This returns an array of EquipmentRequirementAttribute objects.
2. Use an *Index Activity* or *For Each Activity* to get the specific EquipmentRequirementAttribute that you need out of the array. For example: you may check the EquipmentClass property on each EquipmentRequirementAttribute to find which "Mixer" class is specified in the work request.
3. Use the output of the Index or the current value of the For Each to get your equipment name.

As a result, in the Status Message expression, you will need the following:

"Waiting for " + {forEachActivity1}.{CurrentValue}.{Equipment} + " to finish processing."

Related reference

[Visible Property \(on page 1382\)](#)

[Call Method Activity \(on page 1433\)](#)

[Index Activity \(on page 1442\)](#)

[For Each Activity \(on page 1483\)](#)

Configure a Status Message activity

Before you begin



Note:

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:

- At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, select **Update Status**.
 7. In the **Message** field:
 - enter a message.
 - click the ellipsis button, and then in the **Edit String** dialog box, enter a message.
 8. If applicable, to include a bind or expression, click the drop-down arrow.

**Tip:**

For more information, see Property Value Types.

9. Click **Save**.

Related information

[Add a resource to the workflow model \(on page 1395\)](#)

Form Activity

The Form activity allows user-created forms and displays to be exposed in the Task List of a running workflow.

Forms are added and created in the Global Displays model. The Form activity binds to the form that is located in this model. Forms are visible in a user's Task List when a workflow is running. Logic can also be included in a form that allows it to recognize an event occurrence and respond to it. For example, a form can either prompt an operator to perform an action, such as entering information in a form, or it can ensure that an action performed by an operator, such as scanning a bar code, automatically submits the form. It is the event handler in the Form activity that ensures these operations. For more information, see *Event Handlers*.

Electronic signatures can be configured for a form and those signatures are collected at run time and stored in your SQL Server database for audit trailing purposes. For more information, see *Purpose of Electronic Signatures*.

During run time, when a Form activity is reached in a task step, execution pauses until the form is completed by an operator. Forms can be submitted or cancelled by the personnel or location assigned to the subprocess. However, for some types of forms, you can configure them to be automatically submitted or cancelled. You can also choose to allow a user to cancel a form.

For information on forms, see:

- *Forms*, for creating forms in the Designer
- *Form Development in Visual Studio*
- *Form Templates*, for using pre-existing forms



Important:

The *Visible Property* of the containing subprocess must be set for this activity to be valid.

Form Types

A Form activity used in a workflow can either be configured to use a regular, hard-coded form, or it can be dynamic, where the form shown is determined at run time.

Therefore, you can use the Form activity in two ways:

Regular Form

Specify a form or display in the system to be revealed at run time.

Dynamic Form

Bind the form to a directory resource variable in the workflow, allowing the workflow to determine the form to show at run time. The Form activity must specify a template form that describes the metadata (parameters and events) that the run-time form requires.

General Settings

The following table describes the available attributes for the Form activity. To clear any field, select Reset to Default using the arrow next to the field.

Attribute	Value	Description
Form	User-defined	Regular Forms:

Attribute	Value	Description
		<p>To select a specific form or display to be shown at run time, click the ellipsis button to select a resource from the Forms or Displays folder.</p> <p>Dynamic Forms:</p> <p>To allow the workflow to determine the form or display to show at run time, click the arrow to bind the form to a directory resource variable, and then select a template.</p> <div data-bbox="1027 884 1422 1192" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: You must select a template; otherwise, your workflow will receive a validation error at run time. </div>
Template	User-defined	<p>Dynamic Forms:</p> <p>Click the ellipsis button to select the template that defines the input and output parameters. You must select a template when you perform a bind.</p> <div data-bbox="1027 1539 1422 1778" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: The form and template form must have matching parameters, including matching display names; </div>

Attribute	Value	Description
		 otherwise, the running workflow will fail.
Allow user to cancel form	N/A	<p>Select this check box if you want users to have the option of cancelling the form during run time. If this check box is selected, a Cancel button appears at the bottom of the form.</p>
Maintain form record upon completion	N/A	<p>Select this check box if you want to receive completed form data after a form has been submitted or cancelled.</p> <div data-bbox="1027 888 1422 1402" style="border: 1px solid #00a0c0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note: If this box is not checked:</p> <ul style="list-style-type: none"> • No data will be stored in memory. • Completed form information will not be available in Task Step Details or in Task History. </div>
Electronic Signature Settings	N/A	<p>Specifies personnel with the authority to <i>sign and verify forms</i>.</p>
Add Event Handler	N/A	<p>Click this button to add an event handler. An event handler can be used both in regular preconfigured forms or in dynamic forms.</p> <p>For more information, see <i>Event Handlers</i>.</p>

General Inputs and Outputs

The Form activity input and output parameters are based on the selected form or display.

The following table describes the default output properties for this activity.

Output Property	Data Type	Description
Cancelled	Boolean	If the Allow user to cancel form check box is selected, this property value is returned.
E-Signature	String	Returns a record of the completed electronic signature if one was collected for the form activity.

Configure a Form activity

Before you begin

A Form activity used in a workflow can either be configured to use a regular, hard-coded form, or it can be dynamic, where the form shown is determined at run time.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Drag and drop a **Local Subprocess** into the workflow and double-click it, or double-click an existing subprocess.
6. To use the functionality of a form created in the Forms Designer, connect it to a Form activity by dragging and dropping it into the subprocess.
7. In the **Form** field:

To use a...	Do this...
regular form	<ol style="list-style-type: none"> a. Click the ellipsis button to select your preconfigured form from the Universal Browser. b. Click OK.

To use a...	Do this...
dynamic form	<ol style="list-style-type: none"> From the drop-down arrow, select Bind to bind the form to a directory resource variable. In the Template field, click the ellipsis button to select a form template.

- Configure the available input parameters.
- Optional:** Select the **Allow user to cancel form** check box.
- Optional:** Select the **Maintain form record upon completion** check box if you want to view form data after a form has been completed.
- In the **Out Comments** field, configure the available output parameters.
This field reflects the output available to the operator for the selected form.
- If applicable, to clear field contents, click the drop-down arrow, and then select **Reset to Default**.
- To activate the form at run time, click the **Overview** tab, and then click **Enable**.

**Note:**

Only activated subprocesses are available for workflows.

- Click **Save**.

Inputs and Outputs for Web Forms

You can use the **Designer** to create web-based forms. Then you use these forms in a Form activity to make them available to an operator during run time. The input and output parameters based on the selected form or display are shown in a workflow Form activity.

Text

Input Parameters

Inputs	Data Type	Description
Name.Value	String	<p>Specifies the text value for a named text field in a form.</p> <p>You can perform a <i>bind</i>, build an <i>expression</i>, or set a <i>literal</i> value against this parameter.</p>

Image

Input Parameters

Inputs	Data Type	Description
Name.URL	String	<p>Specifies the web address where the image is located.</p> <p>You can perform a <i>bind</i>, build an <i>expression</i>, or set a <i>literal</i> value against this parameter.</p>

List Boxes

Input Parameters



Note:

If both the initial index and value are set, and they both are different, then the index takes precedence over the value.

Inputs	Data Type	Description
Name.Value	String	<p>Specifies the initial selection for a named control by string value. The initial value is the default selection displayed when a form loads. If an invalid value is set, no selection is made.</p> <p>You can perform a <i>bind</i>, build an <i>expression</i>, or set a <i>literal</i> value against this parameter.</p>
Name.InitialIndex	String	<p>Specifies an initial selection for a named control by 0-based index value. The initial value is the default selection displayed when a form loads. If an invalid value is set, no selection is made.</p>

Inputs	Data Type	Description
		You can perform a <i>bind</i> , build an <i>expression</i> , or set a <i>literal</i> value against this parameter.
Name.Label	String	<p>Specifies the label text for a named control in a form.</p> <p>You can perform a <i>bind</i>, build an <i>expression</i>, or set a <i>literal</i> value against this parameter.</p>
Name.OptionNames	String[]	<p>Specifies the complete list of display name options available for this form control. To modify these options, click the ellipsis button to open the String Collection Editor, and then enter other options.</p> <p>You can <i>bind</i> this item to defined variables.</p> <div data-bbox="1024 1142 1419 1409" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: For custom forms, plain text option names only are supported, not arbitrary HTML.</p> </div>
Name.OptionValues	String[]	<p>Specifies the complete list of internal option values available for this form control. To modify these options, click the ellipsis button to open the String Collection Editor, and then enter other options.</p> <p>You can <i>bind</i> this item to defined variables.</p>

Output Parameters

Outputs	Data Type	Description
Name.Value	String	Contains the first user-defined text value for the named control in a form. For example, the output property ListBox001.Value contains the user supplied value for a list box with name of List-Box001 .
Name.Values	String[]	Contains the user-defined text values for the named control in a form. These values are used for the multiple entry selection feature of the list box control. If no selection is made, then this array is empty.
Name.SelectedIndex	Int32	Contains the 0-based index of the first user selection for the named control in a form. If no selection is made, then this value is set to -1.
Name.SelectedIndexes	Int32[]	Contains the 0-based indexes of all user selections for the named control in a form. These values are used for the multiple entry selection feature of the list box control. If no selection is made, then this array is empty.

Radio Buttons

Input Parameters

**Note:**

If both the initial index and value are set, and they both are different, then the index takes precedence over the value.

Inputs	Data Type	Description
Name.Value	String	<p>Specifies the initial selection for a named control by string value. The initial value is the default selection displayed when a form loads. If an invalid value is set, no selection is made.</p> <p>You can perform a <i>bind</i>, build an <i>expression</i>, or set a <i>literal</i> value against this parameter.</p>
Name.InitialIndex	String	<p>Specifies an initial selection for a named control by 0-based index value. The initial value is the default selection displayed when a form loads. If an invalid value is set, no selection is made.</p> <p>You can perform a <i>bind</i>, build an <i>expression</i>, or set a <i>literal</i> value against this parameter.</p>
Name.Label	String	<p>Specifies the label text for a named control in a form.</p> <p>You can perform a <i>bind</i>, build an <i>expression</i>, or set a <i>literal</i> value against this parameter.</p>
Name.OptionNames	String[]	<p>Specifies the complete list of display name options available for this form control. To modify these options, click the ellipsis button to open the String Collec-</p>

Inputs	Data Type	Description
		<p>tion Editor, and then enter other options.</p> <p>You can <i>bind</i> this item to defined variables.</p> <div style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note: For custom forms, plain text option names only are supported, not arbitrary HTML.</p> </div>
Name.OptionValues	String[]	<p>Specifies the complete list of internal option values available for this form control. To modify these options, click the ellipsis button to open the String Collection Editor, and then enter other options.</p> <p>You can <i>bind</i> this item to defined variables.</p>

Output Parameters

Outputs	Data Type	Description
Name.Value	String	<p>Contains the user-defined text value for the named control in a form.</p> <p>For example, the output property Radio001.Value contains the user supplied value for a radio button with name of Radio001.</p>
Name.SeletedIndex	Int32	<p>Contains the 0-based index of the first user selection for the named control in a form. If no selection</p>

Outputs	Data Type	Description
		is made, then this value is set to -1.

Combo Boxes

Input Parameters



Note:

If both the initial index and value are set, and they both are different, then the index takes precedence over the value.

Inputs	Data Type	Description
Name.Value	String	<p>Specifies the initial selection for a named control by string value. The initial value is the default selection displayed when a form loads. If an invalid value is set, no selection is made.</p> <p>You can perform a <i>bind</i>, build an <i>expression</i>, or set a <i>literal</i> value against this parameter.</p>
Name.InitialIndex	String	<p>Specifies an initial selection for a named control by 0-based index value. The initial value is the default selection displayed when a form loads. If an invalid value is set, no selection is made.</p> <p>You can perform a <i>bind</i>, build an <i>expression</i>, or set a <i>literal</i> value against this parameter.</p>
Name.Label	String	Specifies the label text for a named control in a form.

Inputs	Data Type	Description
		You can perform a <i>bind</i> , build an <i>expression</i> , or set a <i>literal</i> value against this parameter.
Name.OptionNames	String[]	<p>Specifies the complete list of display name options available for this form control. To modify these options, click the ellipsis button to open the String Collection Editor, and then enter other options.</p> <p>You can <i>bind</i> this item to defined variables.</p> <div data-bbox="1024 873 1419 1140" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: For custom forms, plain text option names only are supported, not arbitrary HTML.</p> </div>
Name.OptionValues	String[]	<p>Specifies the complete list of internal option values available for this form control. To modify these options, click the ellipsis button to open the String Collection Editor, and then enter other options.</p> <p>▢</p> <p>You can <i>bind</i> this item to defined variables.</p>

Output Parameters

Outputs	Data Type	Description
Name.Value	String	<p>Contains the user-defined text value for the named control in a form.</p> <p>For example, the output property ComboBox001.Value contains the user supplied value for a combo box with name of ComboBox001.</p>
Name.SelectedIndex	Int32	<p>Contains the 0-based index of the first user selection for the named control in a form. If no selection is made, then this value is set to -1.</p>

Check Boxes

Input Parameters

Inputs	Data Type	Description
Name.Value	String	<p>Specifies the initial activation state of a named control in a form.</p> <p>You can perform a <i>bind</i>, build an <i>expression</i>, or set a <i>literal</i> value against this parameter.</p>
Name.Label	String	<p>Specifies the label text for a named control in a form.</p> <p>You can perform a <i>bind</i>, build an <i>expression</i>, or set a <i>literal</i> value against this parameter.</p>

Output Parameters

Outputs	Data Type	Description
Name.Value	Boolean	Contains <i>true</i> or <i>false</i> depending on whether the operator selected or cleared the named control in a form.

Text Area

Input Parameters

Inputs	Data Type	Description
Name.Value	String	Specifies the initial text value for a named control in a form. You can perform a <i>bind</i> , build an <i>expression</i> , or set a <i>literal</i> value against this parameter.
Name.Label	String	Specifies the label text for a named control in a form. You can perform a <i>bind</i> , build an <i>expression</i> , or set a <i>literal</i> value against this parameter.

Output Parameters

Outputs	Data Type	Description
Name.Value	String	For example, the output property TextArea001.Value contains the user supplied value for a text area with name of TextArea001 .

Text Boxes

Input Parameters

Inputs	Data Type	Description
Name.Value	String	Specifies the initial text value for a named control in a form. You can perform a <i>bind</i> , build an <i>expression</i> , or set a <i>literal</i> value against this parameter.
Name.Label	String	Specifies the label text for a named control in a form. You can perform a <i>bind</i> , build an <i>expression</i> , or set a <i>literal</i> value against this parameter.

Output Parameters

Outputs	Data Type	Description
Name.Value	String	Contains the user-defined text value for the named control in a form. For example, the output property TextBox001.Value contains the user supplied value for a text box with name of TextBox001 .

Related reference

[Workflow Binds \(on page 1286\)](#)

[Workflow Expressions \(on page 1286\)](#)

[Workflow Literals \(on page 1289\)](#)

Event Handlers

Event handlers are activity sequences that are triggered by a specific event. They allow a workflow to run logic when the events occur while the form is active.

Two types of events can be selected for a form event handler. One is a form event, which is part of the selected form. The other is a workflow event, which is available from External Events or the Create Event activity.

When a workflow is running, only one event handler can run at a time; event handlers cannot run parallel. The events are held in a queue until another event handler finishes running. A limit of 10 event handlers can be queued during run time. If an event handler faults, the Form activity also faults.

An event handler can also update, as well as write to, input properties of a Form activity. Form authors can specify whether a form input property can be updated inside the form event handler. If these properties cannot be updated, they will not be available to use in a bind. If you use these properties in an expression, you cannot change the values.

Limits

A Form activity has a limit of 25 event handlers, and 15 events that can be defined by a form.

Event Source

The event source is the name of the resource used as the source of the event. Event properties are then generated for the event handler based on this source.

Event Handlers in Dynamic Forms

A Form activity used in a workflow can either be configured to use a regular, hard-coded form, or it can be dynamic, where the form shown is determined at run time.

An event handler can be added for a dynamic form and it acts identically to an event handler that is added for a regular form. For a dynamic form, however, the events defined on the template must be equivalent to those on the run-time form (display names must match); otherwise, the handlers will not run.

Completed Event Handlers

This section allows you to select how you want your form to behave when this event handler completes. You can choose whether the form submits or cancel automatically.

Attribute	Value	Description
Submit	User-defined	If set to True, the form submits when the event handler completes. If set to False, the form re-

Attribute	Value	Description
		remains active when the event handler completes.
Cancel	User-defined	If set to True, the form cancels when the event completes. If set to False, the form remains active when the event handler completes.

Task Step Behavior

This Boolean property determines if the task step is acquired when the event occurs.

Attribute	Value	Description
Acquire	Default=True; User-defined	If set to True, the containing task step will be acquired by the client that raised the event. If set to False, the task step is not automatically acquired by the user during run time.

Event Properties

Event properties are generated for the event handler based on the selected event source. The following table describes the common event properties related to an event source.

Attribute	Data Type	Description
Description	String	Specifies additional information related to the event definition resource.
EventNodeAddress	EventAddress	Specifies the Directory Address of the event that occurred.
ID	GUID	Specifies the unique key that identifies the event definition in the data model.
KeepArchive	Boolean	Specifies whether the event associated with the workflow sched-

Attribute	Data Type	Description
		ule is maintained in history by the event manager.
Name	String	Specifies a display name for the selected property.
Timestamp	DateTime	Specifies the actual UTC time that the event occurred.

Add an event handler to a Form activity

Event handlers are activity sequences that are triggered by a specific event. They allow a workflow to run logic when the events occur while the form is active.

Before you begin



Note:

External Events, a Create Event activity, or a form event must be defined in order to add an event handler.

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow. For more information, see **Add a resource to the workflow model**.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Drag and drop a **Local Subprocess** into the workflow and double-click it, or double-click an existing subprocess.
6. Select the Form activity and then, on the activity configuration panel, click **Add Event Handler**.
The **Select Event** dialog box appears.
7. Click the **Activities**, **Subprocess**, or **Workflow** tab, select an event, and then click **OK**.
8. Click **OK**.

Configure an event handler

Before you begin

**Note:**

External Events, a Create Event activity, or a form event must be defined in order to add an event handler.

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Drag and drop a **Local Subprocess** into the workflow and double-click it, or double-click an existing subprocess.
6. Select the Form activity and then, on the activity configuration panel, click **Add Event Handler**.
The **Select Event** dialog box appears.
7. Click the **Activities, Subprocess, or Workflow** tab, select an event, and then click **OK**.
8. Click **Save**.
9. To modify the event source, on the activity configuration panel, click the ellipsis button, and then select a new event.

**Note:**

You cannot set the same initial input value to a form more than once. You can set the form value back to the initial value by performing these steps:

- a. Write a different value, such as an empty string.
- b. Write the original value again.

10. In the **Task Step Behavior** section, to ensure that the step is acquired when an event occurs, select True. If you do not want this behavior, select False.
11. In the **Form Behavior on Complete** section, to automatically submit or cancel the form, select True in the related field.

**Note:**

You can set only one of these properties to True. Both **Submit** and **Cancel** cannot be True.

12. Click **Save**.

Event Handlers for Web-based Forms

You can raise form events from a custom web-based (HTML) form. From the Form activity's event handler, you can update input parameters and set form behavior, such as submitting and cancelling the form.

Sample HTML Form

To define form events, give your event an alphanumeric name, and then call the *fireFormEvent* method with the name of the event that you want to fire.

**Note:**

Form events in web-based HTML5 forms are supported only in Vision.

```
<!DOCTYPE html>
<html>
  <head data-formevents="MyFormEvent1, MyFormEvent2">
    <title>Custom HTML Form</title>
  </head>
  <body>
    <form name="form" id="myFormId">

      <!-- Form Markup -->

      <button type="button" onclick="fireFormEvent('MyFormEvent1')">Fire MyFormEvent1</button>

      <button type="button" onclick="fireFormEvent('MyFormEvent2')">Fire MyFormEvent1</button>

    </form>
  </body>
</html>
```

Add an event handler to a web-based form

Event handlers are activity sequences that are triggered by a specific event. They allow a workflow to run logic when the events occur while the form is active.

Procedure

1. Write a page (HTML document) that contains the form data that you want to use.
For a basic sample document, see *Event Handlers for Web-based Forms*.
2. In the *data-formevents* attribute, define the form event.
3. In the navigator, click **Global Displays > Forms**.
4. In the **Displays** panel, click **Display Editor**.
5. On the **Overview** tab, select **Use a web page**, and then add the URL that points to where the HTML document is hosted.
6. Click **Save**.
7. In the navigator, click **Workflow > Workflows**.
8. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
9. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
10. Click the **Workflow Editor** tab.
11. Drag and drop a **Local Subprocess** into the workflow and double-click it, or double-click an existing subprocess.
12. Insert a *Form activity*.
13. In the **Form** field, select the form that you created.
14. Configure the event handlers, as required.



Note:

If you want to add more event handlers, you must repeat steps 12 and 13 after updating the HTML document, and then configure the handlers again.

What to do next



Important:

You cannot set the same initial input value to a form more than once. You can set the form value back to the initial value by performing these steps:

1. Write a different value, such as an empty string.
2. Write the original value again.

Related reference

[Event Handlers for Web-based Forms \(on page 1519\)](#)

[Form Activity \(on page 1499\)](#)

Configuring Electronic Signatures

To include a signature with a task/task step action or a form, add personnel with the authority to sign and verify forms. By default, all signatures are configured at the workflow level.

Override Workflow E-Signature Check Box (Subprocess Only)

All signatures are configured at the workflow level unless you override this setting at the subprocess level.

Attribute	Value	Description
Override Workflow E-Signature	User-defined	Select this check box to configure e-signature settings for a task step. When you select this check box, the Electronic Signature Settings area becomes active. For more information on configuring this section, see below.

Signature Required Check Box (Workflow and Subprocess)

If the task/task step action or form for your run-time workflow requires a signature, select the **Signature Required** check box.



Tip:

You can bind these properties to a resource related to a workflow or subprocess.

Attribute	Value	Description
Valid Performers	User-defined	Click the ellipsis button to select a personnel resource who has specific security authorization to perform this action. Members of this group signs forms related to the Form activity in the workflow.
E-Signature Description	User-definedString	Specifies the reason for the signature.

Close if Signature Fails Check Box (Workflow and Subprocess)

If you select this check box, the task/task step action or form will close after the number of failure attempts has been reached. The workflow will continue to execute even if the signature does not work or is invalid.

Verifier Required Check Box (Workflow and Subprocess)

If the first signature (that is, the performer's signature) for your task/task step action or form requires a back-up signature to verify the original signature, then select this check box. If the form does not receive a valid signature, the workflow will not continue to execute.



Tip:

You can bind this property to a resource related to a workflow or subprocess.

Attribute	Value	Description
Valid Verifiers	User-defined	Click the ellipsis button to select a personnel resource who has specific security authorization to perform this action. Members of this group verify the signature(s) on the form after it is submitted.

Configure electronic signatures for use during workflow run time

You can configure electronic signatures to be available when a workflow is running to verify a task/task step action or a form.

Procedure

1. You can configure an electronic signature from various locations within a workflow depending on which action during run time you want to set an electronic signature against.

To add signature to a...	Do this...
Task	<ol style="list-style-type: none"> a. Open a workflow model resource. b. At the workflow level, click the E-Signature tab.

To add signature to a...	Do this...
Task Step	<ol style="list-style-type: none"> a. Open a workflow model resource. b. Drag and drop a Local Subprocess into the workflow and select it, or select an existing subprocess. c. Click the E-Signature tab. d. Select the Override Workflow E-Signature check box.
Form	<i>Configure a Form activity.</i>

2. On the configuration panel, in the **Electronic Signature Settings** area, select the **Signature Required** check box.
3. In the **Valid Performers** field, click the ellipsis button to select the security group (personnel resource) that the signature performer must be a member of.
4. If you want the task/task step action or form to close and the workflow to proceed if the electronic signature fails, select the **Close if Signature Fails** check box.

**Note:**

A task/task step action or form can be cancelled without a signature if it has been preconfigured to do so; however, if you cancel, the task step is skipped and the workflow continues.

5. If a second signature is required before the task/task action or form can be submitted, do the following.
 - a. Select the **Verifier Required** check box.
 - b. In the **Valid Verifiers** field, click the ellipsis button to select the security group (personnel resource) that the signature verifier must be a member of.
6. In the **E-Signature Description** field, enter a description of the action being performed or the values that were entered on the task/task step action or form. This description is recorded with the electronic signature record.
7. Click **Save**.

Examples for Binding Valid Performers and Verifiers

Workflow provides the ability to bind valid performers and valid verifiers to an input parameter or the current value of a data item.

The following examples describe how to use binding for valid performers and verifiers.

I: Bind a valid performer or verifier to an input parameter on a global subprocess

You develop a global subprocess that will be used in multiple places in your facility. In this subprocess you have included a form activity that is configured with electronic signature requirements.

One of the workflows that uses the subprocess controls Line A and the other workflow controls Line B. You want to ensure that a different group of users can sign this form. For example, for Line A, the group Line A Operators must perform the signing; for Line B, the group Line B Operators must perform the signing.

To ensure that the correct group of users can sign the form, configure an input parameter for the subprocess so that the security group DirectoryResource data type is passed into the subprocess. On the form activity, bind the Valid Performers group property to the value of the input parameter.

In this way, you can specify different valid groups based on the workflow the subprocess is being used in without having to rewrite or duplicate your subprocess logic.

II: Bind a valid performer or verifier to the current value of a data item

To use this method, use a Search call method (see *IDirectorySearch* service interface) to dynamically get to a different security group and bind to its result.



Note:

If you follow either of these examples to configure a binding that uses a DirectoryResource data type, you must ensure that your logic is correctly set up so that it resolves to a security group at run time.

Write to File Activity

The Write to File activity allows you create one or more log files for any workflow.

You can configure multiple Write to File activities in a single workflow. Each activity can write to a separate file or you can append the content to an existing file.

Workflow log files are automatically written to the folder `%ProgramData%\Proficy\Workflow Files`. You can include additional folders to the file path that are relative to this path; for example, `temp \<filename>.txt`.

Related reference[Common Properties \(on page 1416\)](#)[Workflow Expressions \(on page 1286\)](#)[Workflow Binds \(on page 1286\)](#)[Workflow Literals \(on page 1289\)](#)**Related information**[Activities Used to Create Workflows \(on page 1416\)](#)

Configure a Write to File activity

Prerequisites[Add a resource to the workflow model \(on page 1395\)](#)**Procedure**

1. In the navigator, click **Workflow > Workflows**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Double-click the step containing the activity you want to work with.
6. In the **Content** field, from the drop-down arrow:

Select...	... and then
Literal	Enter the text that you want to write to the log file.
Bind	Bind the content to a variable.
Expression	Write an expression that will write the appropriate content to the log file.

7. In the **File Path** field, enter a name for the log file.

**Note:**

- Workflow log files are automatically written to the folder %ProgramData%\Proficy\Workflow Files.
- You can include additional folders to the file path that are relative to this path.
- If you enter a non-existent file name, the application creates a new file.

8. From the **Include Timestamp** drop-down list, select *True* to include a time stamp with the content being written; select *False* if you do not want to include a time stamp.
9. In the **Max File Size (MB)** field, enter the maximum size of the log file, in MB. When the maximum file size is reached, a new file is created with the same name, but including a number; for example, `<filename>.1.txt`. Each successive file that is generated in this way will be incremented by one.
10. Click **Save**.

eSOP Activities

These activities are used when working with the eSOP Editor.

Read Task Definition Activity

You can use this activity anywhere within a workflow if you need to read a configuration that is specific for use with eSOP.

Inputs

Processes an eSOP task and returns all of the task's properties.

Inputs	Value	Description
Task Info Extended	Object	Specifies the task information returned by a GetExtendedTaskInfo method call.

Outputs

The following properties are associated with the task being processed.

Inputs	Value	Description
ID	GUID	Specifies a unique identifier for the task being processed.
Name	String	Specifies a unique name assigned to a task.
Description	String	Specifies the description provided for the task being processed.

Inputs	Value	Description
Group	String	Specifies a group name for managing similar tasks. You can sort tasks based on their group. Click the drop-down arrow to select an existing group.
Display Name	String	Specifies a name given to a workflow task for the user to see during run time.
Location Assignment	DirectoryResource	Specifies the assigned location of the task being processed.
Personnel Assignment	DirectoryResource	Specifies the assigned human resource of the task being processed.
Enabled	Boolean	Specifies the activation status of the task being processed. True if enabled, False if disabled.
Priority	Int32	Specifies the importance of the task being processed, as determined during configuration.
Expiration	TimeSpan	Specifies the time that the task being processed is set to expire, as determined during configuration.
Steps	List<TaskStepInfoExtended>	Specifies the task steps associated with the task being processed.
Started Emails	List<Email>	Specifies the emails that will be sent, as determined during configuration, when the task being processed starts.
Expiry Emails	List<Email>	Specifies the emails that will be sent, as determined during con-

Inputs	Value	Description
		figuration, when the task being processed expires.
Config Panel Data	List<ConfigPanelDatumInfo>	Specifies all data provided by custom configuration panels for the task.
Categories	CategoryList	Specifies all the categories related to the task being processed.

Read Task Step Definition Activity

You can use this activity anywhere within a workflow if you need to read configuration that is specific for use with eSOP.

Inputs

Processes an eSOP task step and returns all of the task step's properties.

Inputs	Value	Description
Task	Object	The task information returned by a GetExtendedTaskInfo method call.
Subprocess Id	String	Specifies the unique identifier of the subprocess for which you want step configuration.

Outputs

The following properties are associated with the task step being processed.

Inputs	Value	Description
ID	GUID	Specifies a unique identifier for the task step being processed.
Name	String	Specifies a name assigned to the task step being processed.

Inputs	Value	Description
Display Name	String	Specifies a name given to a workflow task step for the user to see during run time.
Location Assignment	DirectoryResource	Specifies the assigned location of the task step being processed.
Personnel Assignment	DirectoryResource	Specifies the assigned human resource of the task step being processed.
Sequence	Int32	Specifies the order within the workflow that the task step is processed.
Enabled	Boolean	Specifies the activation status of the task step being processed. True if enabled, False if disabled.
Expiration	TimeSpan	Specifies the time that the task step being processed is set to expire, as determined during configuration.
Linked Documents	List<LinkedDocInfo>	Specifies the documents that have been linked to the task step being processed during configuration.
Started Emails	List<Email>	Specifies the emails that will be sent, as determined during configuration, when the task step being processed starts.
Expiry Emails	List<Email>	Specifies the emails that will be sent, as determined during configuration, when the task step being processed expires.
Jumps	List<JumpInfo>	Specifies the jumps associated with the task step being

Inputs	Value	Description
		processed, as determined during configuration.
Config Panel Data	List<ConfigPanelDatumInfo>	Specifies all data provided by custom configuration panels for the task step.
Categories	CategoryList	Specifies all the categories related to the task step being processed.

Schedules

Schedules

Schedules are used to run a workflow when an event occurs and provide more configuration options for specific workflows. However, they are created and executed separately from workflows.

Schedules can be used to override values configured for a workflow, such as task priority, work instructions, assignments, expiration, and linked documents. They also provide the ability to enter literal values for workflow input parameters, as well as bind to variables and event properties.

There are four configuration tabs in Workflow Schedules: details, parameters, task details, and tracking configuration. Each tab, starting with Details, works in succession and inherits the original properties of the task details for the selected workflow.

Example

You have a workflow called Machine Cleaning. This workflow is used to clean a number of different models of mixers. A supervisor selects the workflow, assigns a role, location, and input parameters, and then schedules it for a particular day. The workflow is given context with a role, location, the mixer model to clean, and point in time to execute. When the workflow event and schedule are enabled, the schedule is available to be executed.



Note:

You can configure up to 3000 schedules.

User Interface Attributes for Schedules

The **Overview** tab provides general information about the currently selected schedule.

You can select a workflow to be scheduled, as well as the options related to how the schedule starts. Schedules are executed based on a timed event, conditional event, production event, or a manual trigger.

When you create a schedule, you must completely configure all elements, except Version, on the **Overview** tab. Here, you select the workflow to be scheduled, and then you can either assign an event to trigger the schedule, or you can manually run the schedule.



Note:

Only elements on the **Overview** tab require configuration. The other tabs in schedules are optional.

Schedule Configuration

The following table describes the available options.

Area	Value	Description
Select Workflow: Browse	User-defined	Click to open the Universal Browser to select the workflow you want to schedule. <div data-bbox="1024 1062 1419 1285" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: If you select a disabled workflow, this schedule cannot run. </div>
Schedule by Event: Browse	User-defined	Click to open the Select Event dialog box to select an event to trigger the workflow schedule. This button is available only when a workflow is selected. <div data-bbox="1024 1556 1419 1864" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Tip: For information on configuring scheduled events from Workflow Schedules, see <i>Event Sources</i>. </div>

Area	Value	Description
Schedule by Event: Delete	N/A	Click to remove the selected event. This button is available only when an event is selected.
Manual Trigger: Run Workflow	N/A	Click to manually start the selected workflow. This button is available only after a workflow is selected and the schedule is enabled.
Start Task Filter	N/A	Specifies the location and/or personnel to filter the list of tasks presented when you click the <i>Start Task</i> button.

Enable and disable a schedule

Before you begin



Note:

To edit a schedule, it must be disabled.

Procedure

1. In the navigator, click **Workflow > Schedules**.
2. In the resource type list, select a resource.
3. On the **Overview** tab:

Click...	To...
Enable	activate the schedule
Disable	deactivate the schedule

Add a workflow to a Workflow Schedule

Procedure

1. In the navigator, click **Workflow > Schedules**.
2. Click **Add Schedule**.
3. In the resource type list, select a resource.

4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. In the **Select Workflow** section, click **Browse**.
The **Universal Browser** appears.
6. Select the workflow you want to add to the schedule, and then click **OK**.
7. Click **Save**.

Configure workflow or schedule instance behavior

Procedure

1. Select the applicable resource from one of the following locations:

For...	Do this...
Workflows	In the navigator, click Workflow > Workflows .
Schedules	In the navigator, click Workflow > Schedules .

2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. On the **Overview** tab, in the **Workflow Instance Behavior** section, select the appropriate check box for your workflow.
 - To run more than one of the same workflow at the same time, select Allow multiple instances of this workflow to be running at once.
 - For better workflow performance, select Disable persistence for instances of this workflow.
 - To restart stopped instances, select Automatically restart stopped instances of this definition on server start.
5. Click **Save**.

Start Task Filter for Schedules

By setting the Start Task Filter, you limit a user's ability to start a particular task in the Task List during run time. This feature allows you to filter based on the same criteria as task step assignments. Location (computer or equipment) and/or personnel (user or personnel class) control the tasks that can be started by an operator based on credentials and the computer being used.



Note:

A workflow schedule must be associated with a workflow task in order to start it from the Start Task feature, and this workflow (and schedule) must be enabled, with the **Visible** property set to True.

Schedule Configuration

The following table describes the available options.

Area	Value	Description
Is Schedule Visible in Start Task Window	N/A	Select this check box to ensure that the selected schedule appears in the list of tasks presented when you click the Start Task button.
Set Location	User-defined	<p>Click to select an equipment or computer assignment to be used to filter the list of tasks that can be started by an operator.</p> <div data-bbox="1024 863 1427 1803" style="border: 1px solid #00a0c0; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note:</p> <p>If an equipment or computer resource is selected for the location filter, that location <u>must</u> be part of the user's Task List context for the schedule to appear in the Start Task dialog box. If <u>no</u> location context is set, this schedule will not appear in the list.</p> <p>The schedule will be visible, however, for any location context (including no context) if no location filter is selected, provided the personnel filter matches.</p> </div>

Area	Value	Description
Set Personnel	User-defined	Click to select a personnel assignment to be used to filter the list of tasks that can be started by an operator.
Clear	N/A	Click to remove location or personnel settings.

Filter Scheduled Tasks During Run Time

To limit user and computer access to specific tasks, you can assign specific operators and machines to see the schedule in the **Start Task** dialog.

Before you begin



Note:

A workflow schedule must be associated with a workflow task in order to start it from the **Start Task** dialog box during run time, and this workflow and its associated workflow must be enabled.

Procedure

1. In the navigator, click **Workflow > Schedules**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Overview** tab.
5. In the **Start Task Filter** section, select the **Is schedule visible in Start Task window** check box.
6. You can filter tasks by location and/or personnel.

To filter by...	Click...
equipment	Set Location , select a preconfigured piece of equipment or computer from the Universal Browser , and then click OK .
operators	Set Personnel , select a preconfigured class of people or an individual person from the Universal Browser , and then click OK .

7. **Optional:** To remove a location or personnel setting, click **Clear**.
8. Click **Save**.

Event Sources

Event sources are data items that you can subscribe to.

When you configure the scheduled settings in a **workflow schedule** or the trigger settings of a **condition event**, event source properties are available to define further.



Important:

Available event sources depend on the service providers you have installed.

Event Source	Value	Description
Equipment	User-defined	If no equipment resources with property values have been added in the Equipment model, this event source is not available.
User-defined events	User-defined	If no time or condition event resources have been added in the Event model, this event source is not available.
Production	System-defined	A production event is signaled when one of the following occurs: <ul style="list-style-type: none"> • New Work Request • Work Request Status Changed
Workflow	System-defined	A workflow event is signaled when one of the following occurs: <ul style="list-style-type: none"> • Task Expired • Task Step Expired
GE Historian	User-defined server items	If no server items have been added in the Proficy System model, this event source is not available.

Event Source	Value	Description
		<p>A GE Historian event is signaled when the following occurs:</p> <ul style="list-style-type: none"> • Tag Synchronization Event
Material	System-defined	<p>A material event is signaled when one of the following occurs:</p> <ul style="list-style-type: none"> • Material Lot Deleted • Material Lot HasChildren Changed • Material Lot Location Changed • Material Lot Quantity Changed • Material Lot Status Changed • Material Sublot Deleted • Material Sublot HasChildren Changed • Material Sublot Location Changed • Material Sublot Quantity Changed • Material Sublot Status Changed • New Material Lot Created • New Material Sublot Created
Compliance	System-defined	<p>A compliance event is signaled when the following occurs:</p> <ul style="list-style-type: none"> • Electronic Signature Signing Result
Security	System-defined	<p>A security event is signaled when one of the following occurs:</p>

Event Source	Value	Description
		<ul style="list-style-type: none"> • SSO Authorization Failed (executes when a single sign on user is not configured correctly) • The account is locked out (executes when a user account is locked out) • User Account Created (executes when a new user account is created)

Add or delete an event source to a schedule

Procedure

1. In the navigator, click **Workflow > Schedules**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Overview** tab.
5. In the **Scheduled Settings** section, click **Browse**.
The **Select Event** dialog box appears.
6. Select the event you want to add to the schedule, and then click **OK**.



Tip:

To remove an event from the schedule, click **Delete**.

7. Click **Save**.

Workflow Event Source Properties

Event sources are data items that you can subscribe to when you configure a workflow schedule. You use event sources to define when the selected workflow schedule is triggered

When an event source is added to a workflow schedule, you can bind the parameters from the attached workflow to event properties.

**Note:**

You can only bind parameters to property values of similar data types.

Common Event Source Properties

The following table describes the common properties available for each event source.

Property	Data Type	Description
Id	GUID	Unique key that identifies the event definition in the data model.
Name	String	Display name of the event definition resource.
Description	String	Additional information related to the event definition resource.
Timestamp	DateTime	The actual UTC time that the event occurred.
EventNodeAddress	EventAddress	The Directory Address of the event that occurred.
EventType	String	The .NET type of the event occurrence.
KeepArchive	Boolean	If the event associated with the workflow schedule is maintained in history by the event manager, this is True.

Task Step Expired Event Source

This event is signalled when a task step's expiry time is reached before a task is complete.

Following are all available properties that are specific to this event source.

Property	Data Type	Description
Schedule	DirectoryResource	The schedule for the instance that expired. This value is null

Property	Data Type	Description
		if the workflow was not started from a schedule.
StepDefinitionID	String	The unique identifier of the task step definition that expired. It corresponds to the ID property in subprocess activities.
TaskInstanceID	GUID	The unique identifier of the task instance that expired. It corresponds to the TaskInstanceID property in a workflow.
TaskName	String	A unique name that is distinguishable from the workflow name.
TaskPriority	Nullable Int32	The task priority order for the current workflow
TaskStartTime	DateTime	The date and time that a task started.
TaskStepAssignedLocation	DirectoryResource	The location that a task step is assigned to.
TaskStepAssignedPersonnel	DirectoryResource	The person that a task step is assigned to.
TaskStepCurrentComputer	DirectoryResource	The current computer that a task step is running on.
TaskStepCurrentUser	DirectoryResource	The current user that a task step is assigned to.
TaskStepName	String	A unique name that is distinguishable from the subprocess name.
TaskStepInstanceID	GUID	The unique identifier of the task step instance that expired. It corresponds to the TaskStepIn-

Property	Data Type	Description
		stancelD property in subprocess activities.
TaskStepStartTime	DateTime	The date and time that a task step started.
Workflow	DirectoryResource	The workflow definition for the instance that expired.

Task Expired Event Source

This event is signalled when a task's expiry time is reached before a task is complete.

Following are all available properties that are specific to this event source.

Property	Data Type	Description
Schedule	DirectoryResource	The schedule for the instance that expired. This value is null if the workflow was not started from a schedule.
TaskInstanceID	GUID	The unique identifier of the task instance that expired. It corresponds to the TaskInstanceID property in a workflow.
TaskName	String	A unique name that is distinguishable from the workflow name.
TaskPriority	Nullable Int32	The task priority order for the current workflow
TaskStartTime	DateTime	The date and time that a task started.
Workflow	DirectoryResource	The workflow definition for the instance that expired.

Schedule Overrides During Run Time

The **Task Details** tab allow you to input more schedule information related to the attached workflow task or subprocess task step.

The information is inherited from the selected workflow; however, you can override previous configurations of task/task steps to meet the needs of your workflow schedule.

Task/Task Steps List

This list highlights the current workflow and its subprocesses that are attached to the schedule. Select the workflow or a subprocess to configure it.

Attribute	Value	Description
Workflow	User-defined	Displays the name of the current workflow.
Subprocess	User-defined	Displays the name of the subprocess(es) contained in the current workflow.

Schedule Parameters

The parameters available in Workflow Schedules are the preconfigured input parameters or data items of the selected workflow. If the selected workflow has no parameters, no information displays on this tab.



Note:

Subprocess level parameters and fault handler parameters are not displayed in the workflow schedule parameters list.

Schedule parameters can be set with a static initial value for the parameter, or can be bound to any variable or event property (if an event is attached to the schedule). The values and the types of values available to be configured depend on the selected workflow. A parameter reconfigured in Workflow Schedules is saved as part of the schedule configuration and overrides the workflow parameters.

Schedule Parameter Overrides



Note:

If any parameter types are changed in an attached workflow after that workflow was attached to the schedule, the original workflow parameters displayed in the schedule appear as invalid.

The following table shows the options available when modifying the input parameters:

Attribute	Value	Description
Parameters for Workflow	User-defined	Displays a list of available input parameters for the selected workflow.
Parameter	User-defined	Specifies a value for the selected parameter. Field options vary depending on the parameter's data type.
Data Sources	User-defined	From the drop-down list, select Data Source to open the Configure Data Source window, and then select a resource to bind to.
Event Sources	User-defined	<p>From the drop-down list, select Event Property to open the Event Properties dialog box, and then select a resource to bind to.</p> <div data-bbox="1024 1094 1419 1362" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: This option is available only if an event is attached to the current schedule.</p> </div>



Tip:

The system prevents you from entering an invalid value or selecting an invalid binding.

Reconfigure parameters from Workflow Schedules

About this task



Note:

To configure parameters, a workflow containing parameters must be attached to the current schedule. The type of parameters available to configure depends on the existing data types associated with the parameter. Therefore, the parameter fields vary.

Procedure

1. In the navigator, click **Workflow > Schedules**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Parameters** tab.
5. To reconfigure parameters:
 - a. In a parameter field, enter a value, click the ellipsis button or link, or use the arrow to change your values.
 - b. From the drop-down list, select a resource to bind to.
6. To remove the value, click the arrow and select **Reset to Default**.
7. Click **Save**.

Override Workflow Priorities and Subprocess Names

The **Details** area is where you can set or override the task priority for the workflow selected in the Task/Task Steps list. You can also rename the display name for this workflow's subprocesses.

Attribute	Value	Description
Workflow: Priority	User-defined; Whole number	<p>Select the Use default workflow priority option to retain the current workflow priority.</p> <p>Select the Override workflow priority option to override the default workflow priority, and then enter a new value for the workflow priority.</p>

Attribute	Value	Description
		 Note: Priority is used for sorting and filtering tasks in the Task List; it does not affect the execution of the workflow.
Subprocess: Override	N/A	Select this check box to override the subprocess name.
Subprocess: Override Display Name	User-defined	Specifies a new name for the subprocess. This field is available only when the Override check box is selected.

Override the workflow's priority for a schedule

Procedure

1. In the navigator, click **Workflow > Schedules**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Task Details** tab, and then click the **Details** tab.
5. In the **Task/Task Steps** panel, select the workflow you want to set the priority for.
6. Select **Override workflow priority**, and then enter a value.



Note:

You can enter whole numbers only.

7. Click **Save**.

Override the task step display name

Before you begin



Note:

A resource must already have been added in order to select it from this list.

Procedure

1. In the navigator, click **Workflow > Schedules**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Task Details** tab, and then click the **Details** tab.
5. In the **Task/Task Steps** panel, select the subprocess you want to change the name for.
6. Select the **Override** check box.
7. In the **Override Display Name** field, enter a new name.
8. Click **Save**.

Work Instructions in Schedules

The Work Instructions area is where you can override the previously configured instructions for the task/task step.

Work instructions are useful for the person performing the task or task step. These instructions appear in the Input panel in the Task List and Task Management while a workflow is in a run time state.

Attribute	Value	Description
Override	N/A	Select this check box to update the work instructions.
Work Instructions	User-defined	Displays the original work instructions for the selected workflow task or subprocess task step.
Schedule work instructions workspace	User-defined	Enter new work instructions for the selected task or task step.

Override work instructions for a schedule

Procedure

1. In the navigator, click **Workflow > Schedules**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Task Details** tab, and then click the **Work Instructions** tab.
5. In the **Task/Task Steps** panel, select:
 - The workflow to add different instructions to the task.
 - A subprocess to add different instructions to the task step.

6. Select the **Override** check box.
7. In the new workspace, enter work instructions.

**Tip:**

You can insert information copied from another application, as well as graphics, but it cannot exceed 4 MB.

8. Click **Save**.

Linked Documents in Schedules

In the Linked Documents area, you can attach a document or web site address for the person performing a running workflow task (or a subprocess task step) to reference. These documents can include anything that an operator requires to perform or complete an operation.

Linked documents appear in the Documents panel in the Task List and Task Management while a workflow is in a run-time state. They are embedded in this panel unless your Microsoft Windows configuration does not allow this, and can be in any application format that the user's workstation supports. However, if a document opens outside the linked documents panel, the user is responsible for closing it.

**Note:**

You must have the ability to apply or assign documents to a task.

**Tip:**

When you link a document at the workflow level, that is, to a task, it remains open while the task runs (unless the user closes it or opens another document). All embedded documents close automatically when a task is completed.

Example

If a maintenance operator is performing a maintenance task on a piece of equipment that he has not worked on for some time, the service manual can be an important reference. The service manual can be linked and then viewed for reference while carrying out the maintenance work.

Attribute	Value	Description
Linked Documents list	User-defined	Displays a list of documents that are linked to the workflow/sub-process. They can be arranged by: <ul style="list-style-type: none"> • document name • original vs. override • override vs. original
Add	User-defined	Click this button to add a linked document to the current workflow or subprocess for this schedule.
Delete	User-defined	Click this button to remove a linked document from the current workflow or subprocess for this schedule.

Add a linked document to a schedule

Before you begin



Tip:

- Mobile-sized Task List only: To view Microsoft Office 2007 and later documents inside an Internet Explorer browser window, add a registry patch.
- Web Task List only: To open embedded documents, set the URI for each link to an absolute path, save documents in a folder on a server, and give the folder share permissions.

Procedure

1. In the navigator, click **Workflow > Schedules**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Task Details** tab, and then click the **Linked Documents** tab.
5. In the **Task/Task Steps** panel, select:

- The workflow to add a document to the task.
 - A subprocess to add a document to the task step.
6. Click **Add**.
 7. Click **Browse** to select the file or web site address, or in the **URL/UNC** field, enter the full path to the document, or the web address of the web site.
 8. In the **URL/UNC Text** field, enter a name for the file or web site, or accept the default.

**Note:**

The full file path or web site address must be entered.

9. To view the document or web site, click **Preview**.
10. Click **Save**.

Remove a linked document from a schedule

Procedure

1. In the navigator, click **Workflow > Schedules**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Task Details** tab, and then click the **Linked Documents** tab.
5. Click the **Task Details** tab, and then click the **Linked Documents** tab.
6. In the **Task/Task Steps** panel, select the workflow or subprocess with the linked document you want to remove.
7. In the **Linked Documents** list, select the document you want to remove, and then click **Delete**.
8. Click **Save**.

Assignments in Schedules

The **Assignments** area is where you can override the previously configured location/personnel assignments for workflow tasks or subprocess task steps, depending on the requirements of the schedule. These task/tasks steps can be sent to an assigned computer or operator at that station to complete.

**Note:**

The schedule must contain either a personnel or location assignment, but it can also contain both. When you clear an assignment, the original workflow assignment is not retained.

Attribute	Value	Description
Original Task/Task Step Assignments	System-defined	Displays the original equipment and personnel assignment configurations for the selected task or task step.
Override	N/A	Select this check box to update the schedule.
Set Location	User-defined	Click to select an equipment or computer resource for the selected workflow task or subprocess task step.
Set Personnel	User-defined	Click to select a personnel or personnel class resource for the selected workflow task or subprocess task step.
Clear	N/A	Click to remove location or personnel settings.

Override the assigned location in a schedule

Procedure

1. In the navigator, click **Workflow > Schedules**.
2. In the resource type list, select a resource.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Task Details** tab, and then click the **Assignments** tab.
5. In the **Task/Task Steps** panel, select:
 - The workflow to add a document to the task.
 - A subprocess to add a document to the task step.
6. Select the **Override** check box.
7. Click **Set Location**.
The **Universal Browser** appears.
8. Select a new resource for the schedule, and then click **OK**.

**Tip:**

Click **Clear** to remove the selection.

9. Click **Save**.

Override the assigned personnel in a schedule

Procedure

1. In the navigator, click **Workflow > Schedules**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Task Details** tab, and then click the **Assignments** tab.
6. In the **Task/Task Steps** panel, select:
 - The workflow to add a document to the task.
 - A subprocess to add a document to the task step.
7. Select the **Override** check box.
8. Click **Set Personnel**.
The **Universal Browser** appears.
9. Select a new resource for the schedule, and then click **OK**.

**Note:**

Click **Clear** to remove the selection.

10. Click **Save**.

Expiration in Schedules

You can set an expiration date and/or time for workflow tasks, subprocess task steps, and scheduled tasks.

An expiration is a visual indication about how much time is available to perform a workflow task (or a subprocess task step). Expirations determine the duration of a task or task step when a workflow or a subprocess is in a run-time state.

In a workflow instance, if a task or task step expires, you can still complete it. The entire process is not lost; however, the task or task step is logged as having timed out, and the workflow state changes

and initiates an event that you can hook into. This event can also be used as a means of escalation. If required, you can then run the task or task step again. A running task is not interrupted if a task expires.

Example

During the execution of a task, one of the task steps expires because of an equipment maintenance issue. The task supervisor can, during the execution of the subsequent task steps, go back and complete or restart the expired task step when the employee and equipment are available to complete the task step.

Attribute	Value	Description
Original Task/Task Step Expiration	System-defined	Displays the original expiration type and time configurations for the selected task or task step.
Override	N/A	Select this check box to update the schedule.

Expiration Options



Note:

The Absolute and Relative expiration types allow you to build an expression or perform a bind.

Action	Value	Description
None	System-defined; Default selection	No expiration is set for the selected item.
Absolute	User-defined	Select this option to define an exact date and time when this item expires. In the Duration field, select a date and time.
Relative	User-defined	Select this option to provide a countdown timer to indicate how long this item has to complete. In the Duration field, you can set the number of days, hours, minutes, or seconds for this item to expire.

Action	Value	Description
		 Tip: Insert a condition event to: provide the completion countdown, or ensure a condition is met to expire this item.
Relative: Condition Event	User-defined	Select this check box to configure a condition event for the item's relative time. Click the Configure Event link to configure the conditional event. For information, see <i>Condition Events</i> .

Override an expiration for a schedule

You can override an expiration date and/or time for workflow tasks and/or subprocess task steps when a workflow is running.

Procedure

1. In the navigator, click **Workflow > Schedules**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Task Details** tab, and then click the **Expiration** tab.
6. In the **Task/Task Steps** panel, select:
 - The workflow to add a document to the task.
 - A subprocess to add a document to the task step.
7. Select the **Override** check box.
8. To override the configured expiration:

Select...	To...
None	indicate that there is no expiration.

Select...	To...
Absolute	modify the exact expiry date and time, and then click the drop-down arrow to select a date and time.
Relative	modify the countdown expiration, and then use the arrows to set the number of days, hours, minutes, or seconds until the expiration.

 **Note:**
To include a condition event, select this check box.

9. Click **Save**.

Tracking Data in Schedules

The Tracking Configuration tab allows you to track subprocess or workflow level data and local variables while a workflow is running.

Overrides

The following table is an overview of the tracking configuration options available when you want to override the current workflow settings:



Note:

To view a tracked variable after a workflow has run, enable tracking at both the process (workflow) and subprocess level.

Attribute	Value	Description
Override Default Tracking for this Schedule	N/A	Select this check box to enable the Tracking Settings options.
Enable Tracking	N/A	Select this check box to enable tracking for the schedule.
Enable Tracking: Track at the Process Level Only	N/A	Select this option to track workflow data only.

Attribute	Value	Description
Enable Tracking: Track at the Process and Subprocess Level	N/A	Select this option to track both workflow and subprocess data.
Workflow/Subprocess	N/A	Select a workflow or subprocess to track its local variables.
Trackable Variables	N/A	Select the local variables you want to track for this schedule.

Override the tracking configuration in schedules

Procedure

1. In the navigator, click **Workflow > Schedules**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Tracking Configuration** tab.
6. Select the **Override default tracking for this schedule** check box.
7. Select the **Enable Tracking** check box, and then select one of the following options:
 - **Track at the process level only** for workflow only variable tracking, or
 - **Track at the process and subprocess level** for tracking at all levels.
8. From the Workflow/Subprocess panel, select the workflow or subprocess that contains the variables you want to track.
9. In the **Trackable Variables** panel, select the check box for each variable you want to track.
10. Click **Save**.

Faults

Faults

Faults are errors or failures that occur within a workflow while it is running.

Faults can occur for any number of reasons, such as issues in activity execution or with one of the runtime services. When a fault occurs, it is raised to the workflow runtime and the workflow is terminated. If a fault is raised while an activity is executing, the remainder of the activity's work is cancelled. If more activities follow the faulted activity, that work cannot be started or completed. With no way to handle these faults, the workflow can never complete.

Faults can be both user-defined and created automatically by the system. A workflow can be configured to recover from faults; however, if a fault is not handled, the workflow ends and logs a description of the fault.

Workflow allows you to work with faults in three ways: through Fault activity, Faulty Scope activity, and Fault Handlers. Each one of these functions has both a versatile and specific role to play when designing a process. By using these functions, expected issues can be resolved without causing a major system disruption.

**Note:**

You can configure up to 1000 faults.

Fault Configuration

Fault configuration provides general information about the currently selected fault.

Attributes

The `Fault Message` specifies a message that will be reported when a fault occurs.

Create a fault resource

Configuring a fault allows you to assign a message to a user-defined fault in order to provide more information about the issue.

Before you begin

**Note:**

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. In the **Fault Message** field, enter the message you want displayed when the fault occurs.
6. Click **Save**.

Use a preconfigured fault resource in a subprocess

Before you begin

You can add only one preconfigured fault for each fault activity. You must also ensure that there is at least one activity following the fault activity.



Note:

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with



Note:

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Select Fault**.
The **Universal Browser** appears.
7. Select the fault you want to add to the fault activity, and then click **OK**.
8. Click **Save**.
9. Test the workflow and view the results. The workflow terminates with the message specified in the fault resource.

Fault Activity

Fault activities allow you to end a workflow in a predictable way under specific conditions. It is used to create a user-defined stop point within a workflow where an error is raised.

If a fault resource is selected when the fault activity is configured, that fault is raised with its associated error message. If a fault has not been selected, the activity raises a default fault, or a non-specific error.

The common use for a fault activity is within a fault scope where the end result is expected and defined. Fault activities should not be used in a linear process as they will cause the process and the workflow to end and begin the postsubprocess and unload steps.

Fault Edit

You can create a fault activity while developing a workflow. You can edit, view, and add a fault activity to any step within this workflow.

The following option is available to configure the fault activity:

Option	Value	Description
Select Fault	User-defined	Click this option to add a fault to a fault activity to raise a single fault. The selected fault displays in the activity configuration panel.

Add a fault activity to a subprocess

You can define a fault that is raised at any specified point in a workflow. You can add only one fault resource for each fault activity.

Procedure

- To add a fault activity, see *Insert an activity*.
- To edit a fault's attributes, see *Edit activity and subprocess attributes*.

Fault Scope Activity

The Fault Scope activity is used to contain activities in a subprocess that are likely to raise errors. If a fault is somewhat predictable, the Fault Scope activity can invoke preset resolutions that deal with the fault and permit the workflow to continue at the point immediately following the scope.

The benefit of fault scopes is that they have built-in resolutions to these failures. Scopes have fault handlers that deal with these issues so that the workflow instance can resume after the scope runs if an error occurs. So, if the fault is handled, the workflow continues at the point immediately following the scope. There can be many levels of scopes, which are nested in the subprocess.

The fault scope activity allows a series of activities to be contained in the scope. If a failure during a workflow instance is predictable, you can add a *fault handler* that is associated with the fault to resolve these circumstances. For example, if you can predict that a new process designed on a production floor may fail some of the time, a special set of fault handlers can be assigned to this scope to deal with the failure.

The fault scope activity is also associated with fault handlers as they pertain to level promotion. Within a fault scope, there are a series of fault handlers at various levels to try and catch any error or failure that is occurring. The fault scope searches for handlers to run. If the scope cannot find a handler to deal with the fault (that is, if no resolution to this fault is found), the fault is promoted to the next level of fault scope where it could possibly be handled by its fault handlers. The next level could be another fault scope activity (if the scope is nested), or upwards to the subprocess/step itself. The same sequence of events happens until there are no further levels to promote to. If there are no more levels (that is, the fault reaches the workflow level), the workflow stops and an error is logged.

**Note:**

You cannot configure a fault scope at the workflow level.

Add a fault scope activity to a subprocess

When you add a fault scope to a step in a workflow, you can define the fault handlers. To view the fault handlers, click **Show fault handlers for this activity**. To return to the normal activity flow for the fault scope, click **Show activity sequence**.

Procedure

- To add a fault activity, see *Insert an activity*.
- To edit a fault's attributes, see *Edit activity and subprocess attributes*.

Fault Handler

The fault handler's main role is to provide a means of recovery from an error in a workflow. It associates a specific error with the recovery process.

One or more fault handlers can be added to a scope. The scope itself is a subprocess or a step within a workflow or a fault scope activity. However, no matter how many fault handlers are within a scope,

only one will run based on the configuration of the handlers. Fault handlers can be configured to handle specific faults, system faults, or all faults. The handler that runs is the first one that is appropriately configured to resolve the fault. The path that is followed to find the appropriately configured fault handler is a linear path (from left to right).

If any handler runs, the currently running subprocess ends and the rest of the workflow continues to run as prescribed, starting at the next subprocess. If no handler runs or an unhandled fault occurs, the current subprocess ends at the fault and the workflow instance ends in the normal way; that is, the postsubprocess, postprocess, and unload steps run before the instance completes.

Fault handlers exist at two levels: within a process step and within a fault scope activity. Logic for the fault handlers can be defined at both levels. It is important to have handlers at both these levels to be able to try and resolve any issues that cannot be resolved at the activity level.

Fault Handler Activity

Fault handler activities are added to a fault handler collection, which are added to fault scope activities. If a fault handler activity is set to handle a certain fault, it will drill down into the handler and run the activities of that handler when that fault occurs. You can edit the handler's attributes at the fault handler level.

The following options are available to configure fault handler activities:

Attribute	Value	Description
Handle All Faults	User-defined	Select this check box to automatically handle any faults that the handler was not configured to resolve. Typically, this option is selected for the last fault handler activity in a series to ensure that all faults are found and resolved. No other options are available when this option is selected.
Handle System Exception	User and system-defined	Select this check box to handle any system fault that was not created by a user. When you select this check box, the Handled Faults Edit link is enabled.

Attribute	Value	Description
Handled User Faults: Select Faults	User-defined	Click this link to open the Universal Browser to select one or more faults to add to the selected fault handler activity. The selected faults display below Handled Faults in the activity configuration panel.
Handled User Faults: Clear List	User-defined	Click this link to remove any user-defined faults previously added to this fault handler activity.

Properties

The following fault handler activity properties are available through these binds: *Write activity* and *Local Variables*.

Attribute	Value	Description
FaultName	System-defined	This provides more information about the fault that occurred.
FaultMessage	System-defined	This provides more information about the fault that occurred.

Related reference

[Write Activity \(on page 1447\)](#)

[Local Variables \(on page 1345\)](#)

Add a fault handler to a fault scope activity

Before you begin



Note:

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the Navigator, click **Workflow > Workflow**.
2. In the Workflows list, select the resource you want to work with



Note:

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. Double-click the process step you want to work with.
7. Insert a fault scope activity.
8. On the activity configuration panel, click **Show fault handlers for this activity** next to **View**.
9. In the fault scope activity you added, click **Click here to add a fault handler**.
10. To add another fault handler activity, click the activity, and then click the plus sign.

Add a fault handler to a process step

Before you begin



Note:

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the Navigator, click **Workflow > Workflow**.
2. In the Workflows list, select the resource you want to work with.

**Note:**

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. Double-click the process step you want to work with.
 7. On the activity configuration panel, click **Show fault handlers for this activity** next to **View**.
 8. In the workspace, click **Click here to add a fault handler**.
 9. To add another fault handler activity, click the activity, and then click the plus sign.

Add an activity to a fault handler activity

Before you begin

**Note:**

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the Navigator, click **Workflow > Workflow**.
2. In the Workflows list, select the resource you want to work with

**Note:**

A resource must already have been added in order to select it from this list. For more information, see .

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
 4. Click the **Workflow Editor** tab.
 5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
-  **Note:**
Activities can exist only between the presubprocess and postsubprocess.
- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
 6. Double-click the process step you want to work with.
 7. On the activity configuration panel, click **Show fault handlers for this activity** next to **View**.
 8. In the workspace, click **Click here to add a fault handler**.
 9. Double-click the fault handler activity you want to add an activity to.
 10. Insert an activity.
 11. To return to the process step, click the link in the breadcrumbs.

Test Workflows

Testing Workflows

You can test and review workflows before they become part of your production environment by performing basic tests and verifying instance control.

If a fault occurs that causes the workflow to terminate, the termination reason displays.



Note:

Testing and debugging workflows before putting them into production is strongly encouraged to prevent faults in a running workflow and to troubleshoot potential issues.

Test Tab User Interface

You can start a test workflow by using various test controls.

The following table describes the available options.

Attribute	Value	Description
Set Workflow Input Parameters	User-defined	Allows you to re-configure existing parameters for the current test.
Test Results	User-defined	Displays the results from a tested workflow.
Test Workflow: Test	User-defined	Click this button to test the current workflow.
Test Workflow: Stop Testing	User-defined	Click this button to stop testing the current workflow.
Test Instance Control: Restart	User-defined	Click this button to restart a running workflow.
Test Instance Control: Stop	User-defined	Click this button to stop a workflow during execution.
Debugging Control: Attach	Workflow-defined	The debugger is automatically attached when you open the Workflow Editor . If you previously detached the debugger, click this button to reattach it.
Debugging Control: Detach	Workflow-defined	If you want to manually detach the debugger, click this button.

Test Parameters

You can configure the parameters of the workflow you are testing during the testing process.

The input parameters and test results sections display information only during a manual test of the current workflow. Test input and output parameters do not remain when you navigate from one workflow to another, and do not remain part of the workflow definition when the workflow is saved.

When you perform a test, you can reconfigure input parameters that are applied during this test only. When the test is complete, you can view the output parameters based on the results of the test. The values available for configuration are based on the parameters that have already been added in other configuration panels.

Related reference[Parameters \(on page 1343\)](#)[Task Configuration \(on page 1313\)](#)**Related information**[Debugging \(on page 1566\)](#)

Configure parameters in a test workflow

Before you begin**Note:**

Parameters must have already been added to a workflow to configure them in a running workflow. For more information, see *Add attributes to a workflow or subprocess*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Test/Debug** tab.
5. In a **Set Workflow Input Parameters** field, enter a value.
6. To remove the value, click the arrow and select **Reset to Default**.

Debug Workflows

Debugging

The debugging function allows you to locate and isolate an error in your workflow in order to fix it before the workflow is used in a production environment.

In the **Workflow Editor**, you can review and assess a workflow to ensure it is ready for use. This is done by attaching the debugger to a workflow and testing it. You use the commands on the Debugger Control Bar to perform these operations.

A debugging session occurs when you attach a debugger to a running workflow, which can be done in both a test or a production environment. If you attach the debugger before the workflow instance runs, the debugger pauses at the load step. At this point, you can set any required breakpoints and then move through the workflow step by step, pausing the workflow at each step or activity before it starts to run. If

the workflow configuration at any of these steps requires outside interaction, such as the Form activity and the Manual start function, you must perform these actions before continuing the debugging session.

Also, at each step, you can see the properties in the Variable Viewer. Each step in a workflow is made up of a variety of configurations. When you are in a debugging session, you can view the parameters, properties, and variables that are associated with both the selected step and the workflow as a whole.

A debugging session finishes when the debugger toolbar is detached or the workflow instance is completed or ended. You can detach the debugger from a workflow instance so that it can continue running normally, or be paused or ended.

When you are in a debugging session, the workflow is in a read-only state. Only one user can attach the debugger to a particular workflow instance at a time; however, it is possible to have multiple users attach the debugger to different workflow instances at the same time. Only five debugging sessions for any given workflow are allowed at one time.

Breakpoints

Breakpoints allow you to intentionally pause a workflow when it reaches a specific location, such as a process step or activity. The workflow automatically pauses whenever a breakpoint is reached if the workflow debugger is attached.

When a workflow instance is paused, you can use the Variable Viewer to inspect the current values of activity properties. This allows you to verify the expected behavior of your workflow and to diagnose any problems. The Variable Viewer not display information while an activity is running; it is available only when the workflow instance is paused.

Breakpoints are managed by a user who has specific security permissions. When the breakpoints are set in a workflow definition, they remain part of the workflow during the current client session. When this session ends, the set breakpoints are removed. You have the option to add and remove a single breakpoint at any process step or activity, or to remove all breakpoints at one time.

Workflow Debug View

You can view a workflow and the value of its variables during a debugging session.

In the workflow Workspace, debugging commands and controls are active on the debugger control bar. This view also shows current workflow activity and active breakpoints. You cannot edit the workflow because the activity panel is not available; however, the workflow Variable Viewer shows the name, current value, and data type of properties, parameters, and local variables within a selected step.

Testing and Debugging Commands

When you attach a debugger to a running workflow, you can perform debugging operations, as well as start and stop a test workflow.

These commands represent the functionality of the buttons on the Debugger Control Bar, as well as the meaning of the images displayed during a debug session.

Icon	Command	This icon...
	Start/Continue	Click this button to: <ul style="list-style-type: none"> • start a new test, or • proceed when a running workflow is paused.
	Break	Click to pause a running workflow at the beginning of the next process step or activity.
	Stop	Click to stop testing the current workflow.
	Step	Click to move through a workflow step by step.
	Add a Breakpoint	Click to add a breakpoint prior to running a workflow to ensure the workflow pauses immediately before the specified activity runs.
	Remove a Breakpoint	Click to remove a specific breakpoint that was set before running the workflow.
	Remove all Breakpoints	Click to remove all breakpoints that were set before running the workflow.
	Activity running	Indicates that there is an activity running inside the highlighted subprocess.

Icon	Command	This icon...
	Activity in progress	Indicates that pause has been clicked on the Debugger Control Bar and that the current activity will continue running until it completes. The break will occur at the next activity, which is indicated by a yellow arrow.
	Activity paused	Indicates the activity that is running.

Pause and resume a workflow in a debugging session

Before you begin



Note:

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. To resume a running workflow, from the Debugger Control Bar, click **Continue**.

Add and remove breakpoints

Before you begin

You can set breakpoints when the debugger toolbar is attached. Breakpoints are indicated by a red dot.

Procedure

1. Start a debugging session. See *Start and end a test debugging session*.



Note:

To start a debugging session for a running workflow, see *View a workflow instance*.

2. On the **Workflow Editor** tab, select a step within the workflow, and then click **Add a Breakpoint**.
3. To remove a single breakpoint, select the step with the breakpoint you want to remove, and then click **Remove a Breakpoint**.
4. To remove all breakpoints, click **Remove all Breakpoints**.

Step through a workflow

Before you begin

When you access your workflow in the Workspace, you can move through it step by step.



Note:

If any workflow configurations require outside interaction, you must perform these actions by logging into a separate Client application before continuing the debugging session. Otherwise, when you navigate away from the debugging session, the session is lost.

Procedure

1. Start a debugging session. See *Start and end a test debugging session*.



Note:

To start a debugging session for a running workflow, see *View a workflow instance*.

2. Select the step you want to start at.
3. From the Debugger Control Bar, click **Step**.
4. To step through each activity, click **Step**.



Note:

Each time you click **Step**, the next activity runs, and then pauses before running the next step or activity. If there are multiple subprocesses, you step through the presubprocess and postsubprocess before and after each subprocess until the final subprocess runs.

View the current values for an activity's properties

Procedure

1. Start a debugging session. See *Start and end a test debugging session*.

**Note:**

To start a debugging session for a running workflow, see *View a workflow instance*.

2. To see the values for each process step or activity, select that step or activity in the workflow. The values display in a table in the Variable Viewer area.

View a workflow instance

Before you begin

You can view the workflow that is currently running. A resource must already have been added in order to select it from this list.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the **Displays** panel, click **Workflow Management > Workflow Editor**.

Start and end a test debugging session

You can troubleshoot problems that may be occurring in a workflow by tracking the flow of logic in a workflow.

Before you begin

**Note:**

A resource must already have been added in order to select it from this list.

Procedure

1. In the navigator, click **Workflow**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Test/Debug** tab.
6. To start a test debugging session, click **Test**.
7. To see the workflow in a debugging session, click the **Workflow Editor** tab.

8. To end a test debugging session, click the:
 - **Test/Debug** tab, and then click **Stop Testing**, or
 - **Stop Testing** button on the Debugger Control Bar.
9. To discontinue the debugging session, click the **Test/Debug** tab.
10. Optionally, to remove the debugger, click **Detach**.

Start a debugging session for a workflow instance

Before you begin

You can view the workflow that is currently running. A resource must already have been added in order to select it from this list.

Procedure

1. In the navigator, click **Workflow**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management Workflow Editor**.
4. From the Workspace, select the workflow instance you want to work with, and then click **Attach**.
The workflow instance opens in the **Workflow Editor** with the debugger attached.

Results



Note:

The debugger detaches automatically if the workflow instance finishes or is stopped.

Kill a debugging session

Before you begin

You can view the workflow that is currently running. A resource must already have been added in order to select it from this list.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.

4. From the Workspace, select the workflow instance you want to end, and then click **Kill Debug Session**.

**Important:**

Authorized users can end a workflow debugging session; however, if the workflow was running before the debug session started, it will continue running after the session ends.

Track Data

Tracking Data in Workflows

The **Tracking Configuration** tab allows you to track subprocess or workflow level data and local variables while a workflow is running.

Tracking Settings

When you enable tracking settings, you can track data at the workflow level, or at both the workflow and subprocess levels. These settings control whether workflow and subprocess data is maintained for reporting purposes, as well as whether you can view this data in Task History in the Task List.

**Note:**

To view a tracked variable after a workflow has run, enable tracking at both the process (workflow) and subprocess level.

Tracked Variables

Tracked variables are the local variables added and configured in the **Workflow Editor**, where you can define them as trackable. They are the only types of preconfigured variables that can be tracked in tracking configuration. These variables can be tracked at both the workflow and subprocess levels. They are viewed in the **Trackable Variables** panel, and are based on the workflow or subprocess you select in their respective configuration panels. From this list, you can further define the variables you want tracked. Depending on the data you want to retain when a workflow is executed, different variables can be selected at different times.

**Tip:**

Tracking configurations can be overridden in a workflow schedule.

Example

Use a Form activity to write to local variables, and then track these variables to ensure the action was completed. When tracking is enabled, the results from the form can be viewed in a report.

**Note:**

Local variables must have already been added to the workflow or subprocess in order to configure them in tracking.

Enable tracking

Before you begin

When you enable tracking, a variety of workflow tracking data is saved for reporting purposes and task history details. For more information on the data that is saved, see *Workflow Tables* in the Reporting Database section.

**Note:**

To view a tracked variable after a workflow has run, enable tracking at both the process (workflow) and subprocess level.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Tracking Configuration** tab.
6. Select **Enable Tracking**.
7. Select a tracking level:
 - To track workflow data only, select **Track at the Process Level Only**.
 - To track at both workflow and subprocess data, select **Track at the Process and Subprocess Level**.
8. Click **Save**.

Track local variables

Before you begin

**Note:**

To view a tracked variable after a workflow has run, enable tracking at both the process (workflow) and subprocess level.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Tracking Configuration** tab.
6. Select **Enable Tracking**.
7. Select a tracking level:
 - To track workflow data only, select **Track at the Process Level Only**.
 - To track at both workflow and subprocess data, select **Track at the Process and Subprocess Level**.
8. From the Workflow/Subprocess panel, select the workflow or subprocess that contains the variables you want to track.
9. In the Trackable Variables panel, select the check box for each variable you want to track.
10. Click **Save**.

Modify tracked local variables

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Tracking Configuration** tab.
6. From the Workflow/Subprocess panel, select the workflow or subprocess that contains the variables you want to track.
7. In the Trackable Variables panel, select or clear variable check box(es).
8. Click **Save**.

Web Services Service Provider

Web Services Service Provider (WSSP)

The Web Services Service Provider extends Workflow capabilities by allowing you to use SOAP (Simple Object Access Protocol) web services that provide WSDL (Web Services Description Language) metadata.

**Important:**

The Call Web Method that was provided in earlier versions of Workflow is still available. You can continue to use it for your legacy web services; however, it is recommended that for all new web services configurations, you should use the Web Services Service Provider to add and configure web service definitions.

The Web Services Service Provider defines the methods and displays used to add and remove web services to and from Workflow, and to configure the security, location URL, and endpoint information used to connect to the web service. It is responsible for publishing information about the web services and their methods to the Directory of Resources so that they can be accessed in the navigator and the method selection dialog boxes (that is, for call methods and forms).

You can add a web service by associating the Web Services Service Provider with a WSDL file located at a specified URL. The WSDL file is downloaded and read to determine the configuration of the web service. The Web Services Service Provider then publishes methods that correspond to the methods described in the web service.

**Note:**

You can create up to 100 web services definitions in your system.

After adding and configuring your web service, you can use the web service by configuring its methods in the existing Workflow call method activity, or by configuring the methods as server methods in forms.

The published web service methods are available in the **Configure Method** dialog box and are listed under an interface that corresponds with the web service definition they are associated with. For example, if you create a web service definition named WebService1, then the interfaces configured for the associated WSDL appear in the **Configure Method** dialog box with the methods listed below it, as WebService1_<contract of service>. If you use the methods with the call method activity, the input and output parameters that were configured for each method are displayed in the activity configuration panel. If you use the methods as server methods in WPF or Silverlight forms, the input and output parameters are displayed in the **Properties** panel of the forms designer.

If the data types configured in the web service are not compatible with Silverlight (for example, Datatable), then the collection types generated in the methods appear formatted as arrays.

Web Service Security

When a web service is added, endpoint information is automatically generated based on the service configuration. If the WSDL file provides multiple endpoints, then in the **Web Service Editor** you can select the one you want to use. If the web service uses a user name/password authentication scheme, then the **User Name** and **Password** fields are available on the **Overview** tab of the **Web Service Editor** where you can enter the optional user credentials. If the web service uses a client certificate authentication scheme, then the **Client Certificate Details** fields are available on the **Overview** tab of the **Web Service Editor**.

The security/authentication mode used by the server determines the binding required by the Workflow client to create a connection. Unsecured web services use the BasicHTTPBinding; all other security modes use the WSHttpBinding. The supported security modes used by both binding types are:

- None
- Message
- Transport

Add a web services definition

A web service definition must point to a valid WSDL file that contains all of the endpoints and methods you want to use for the web service. You can create multiple web service definitions, each pointing to a different WSDL URL.

Before you begin

You must have a web service with valid WSDL running before you can add a web service in Workflow using the Web Services Service Provider.

In order to manage web services, you must be a member of a personnel group that has the *Manage Web Service* key set permission assigned to it.

Web service complex types can be instantiated and initialized using the Code activity. A namespace related to a web service has to be added in the **Namespace Browser** in the **Workflow Editor** when configuring a workflow.

About this task



Note:

The Workflow application server must have the Silverlight SDK 5.0 and the Silverlight Runtime 5.0 installed. Client machines must have Silverlight Runtime 5.0 installed.

Procedure

1. In the navigator, click **Proficy System > Proficy System > Web Services**.
2. Click **Add Web Service**.
3. In the **Name** field, enter a unique name for the web service.
Namespace details are automatically populated in the Namespace field.
4. In the **WSDL URL** field, enter the address of the WSDL file you want to associate with the web service.
5. In the **Description** field, enter more information related to the web service.
6. To change the preconfigured string in the **Namespace** field, enter a new namespace using C# namespace guidelines.
The appropriate namespace prefix is `Proficy.WebServices..`
7. Click **OK**.



Note:

If the web service interfaces and methods do not appear in the Silverlight forms editor, then you must restart the Workflow client.

Configure endpoints for a web service

Endpoints are used to connect to a web service.

Before you begin

In order to manage web services, you must be a member of a personnel group that has the Manage Web Service key set permission assigned to it.

About this task



Important:

If the web service is not licensed (as indicated by the message in the **License Overview** area), then you cannot execute any of the methods contained in the WSDL file.

Procedure

1. In the navigator, click **Proficy System > Proficy System > Web Services**.
2. From the **Web Service Definition** list, select the web service you want to work with.
3. In the **Displays** panel, click **Web Service Editor**.
4. Click the **Overview** tab.
5. In the **Endpoint** field, select an endpoint you want the web service to use for communications.
6. In the **Location URL** field, change the address of the WSDL file only if one of the following has occurred:
 - The web service has been moved to a different server.
 - The server is renamed.
 - The web service is hosted at a different URL.
7. Depending on the authentication scheme the web service uses, do one of the following:

If the authentication scheme uses...	Then, do this...
User Credentials	<ol style="list-style-type: none"> a. In the User Name field, enter a valid user name. b. In the Password field, enter the password for the user.
Certificate Credentials	<ol style="list-style-type: none"> a. In the Store Location field, select the location of the X.509 certificate store you want to use. b. In the Store Name field, select the name of the X.509 certificate store you want to use. c. In the X509 Find Type field, enter the value you want the X509Certificate2Collection.Find method to search for. d. In the X509 Find Value field, select the value you want the X509Certificate2Collection.Find method to search for.



Note:

If the endpoint is configured to use the user name and password details, then in order to test or use this endpoint, you must enter valid information in these fields.

8. Repeat steps 5-6 for each endpoint to configure for the web service.
9. Click **Save**.

Customize send timeout settings for a web services call

You can change the *send timeout* setting for your web service.

Prerequisites

[Add a web services definition \(on page 1577\)](#)

Procedure

1. From the `Proficy Services` folder in your installation directory, `<installdir>\Program Files\Proficy\Proficy Workflow\Program`, open the related web service `ClientConfig` file.
2. In `<basicHttpBinding>`, locate the related binding name, edit the binding by adding the **sendTimeout** attribute, and then set the appropriate time.
For example, `sendTimeout="00:02:00"`.
3. Click **Save**.

Delete a web service definition

If your web service definition is not being used by a running workflow instance, you can delete it.

About this task

You can delete a web service that is configured for use in a non-running workflow; however, when that workflow is loaded in the Workflow Editor, a validation error is displayed indicating that the workflow uses a connection that no longer exists. You cannot delete a web service definition that is being used in a running workflow instance.

If any Silverlight form is configured to use a server method that is associated with a deleted web service, then an error message is displayed in the Feedback Zone. If the same server method contains a complex data type, you cannot configure that method again; the method must be deleted from the form, added again, and then configured.

Procedure

1. In the navigator, click **Proficy System > Proficy System > Web Services**.
2. From the **Web Service Definition** list, select the web service you want to remove.
3. Click **Delete**, and then click **OK**.

Test the web service methods

You can test the methods contained in the web service's WSDL file before you use them in a workflow call method activity or in a form.

Before you begin

In order to manage web services, you must be a member of a personnel group that has the Manage Web Service key set permission assigned to it.

Procedure

1. In the navigator, click **Proficy System Proficy System Web Services**.
2. From the **Web Service Definition** list, select the web service you want to work with.
3. In the **Displays** panel, click **Web Service Editor**.
4. Click the **Test** tab.
5. In the **Method** field, select the method you want to test.
The endpoint associated with the selected method is displayed in the **Method Endpoint** field and the input parameters for the selected method are displayed in the **Inputs** panel.
6. In the **Inputs** panel, where applicable, enter values for the displayed input parameters.
7. Click **Test**.

If the test passes, the results are displayed in the **Outputs** panel. If the test fails, an error is displayed in the Feedback Zone.



Note:

For testing purposes, you can use only simple input such as strings and integers. More complex data types are not supported for testing. This restriction also applies to the output values.

Refresh the web service definition

If your web service definition has changed, you can refresh it.

Before you begin

In order to manage web services, you must be a member of a personnel group that has the Manage Web Service key set permission assigned to it.

About this task

The web service must be refreshed if one of the following has occurred:

- The web service has been moved to a different server.
- The server is renamed.

- The web service is hosted at a different URL.
- The web service metadata has changed (for example, the Service Contract, Operation Contract, or the Data Contracts).

Procedure

1. In the navigator, click **Proficy System > Proficy System > Web Services**.
2. From the **Web Service Definition** list, select the web service you want to work with.
3. In the **Displays** panel, click **Web Service Editor**.
4. Click the **Refresh** tab.
5. In the **WSDL URL** field, enter the address of the WSDL file you want to associate with the web service.
6. Click **Refresh**.
7. Restart the Workflow client and application server to apply the changes.



Note:

If the WSDL URL has changed, the new value can be saved only if the refresh is successful; otherwise, it reverts to the old value.

Endpoint Attributes for a Web Services Service Provider

You can configure the endpoints to use for the current web service.

The following table describes the information available for configuring a web service on the **Overview** tab.

Property	Value	Description
License Overview	System-defined	Displays a message pertaining to the licensing status of the current web service.
Endpoint	User-defined	Specifies the endpoint name from the generated client configuration file. Endpoints are associated with an address (location URL), binding, and contract.
Location URL	User-defined	Specifies the address of the service used to communicate with the web service.

Property	Value	Description
User Name	User-defined	Specifies a user name that is valid within the web service's user name/password schema.
Password	User-defined	Specifies the user's password for the web service's user name/password schema.
Store Location	System-defined	Specifies the location of the x.509 certificate store.
Store Name	System-defined	Specifies the name of the X.509 certificate store to open.
X509 Find Type	System-defined	Specifies the type of value the X509Certificate2Collection.Find method searches for.
X509 Find Value	User-defined	Specifies the value the X509Certificate2Collection.Find method searches for.
Web Service Methods	N/A	Displays a list of the methods published from the web service. You can expand each of the methods to view the input and output parameters.

Web Service Test

The Test tab is used to execute the methods published by the web service. This allows you to determine whether the methods produce results as expected before using them in a call method or form.

The following table describes the information available for testing a web service.

Property	Value	Description
Method	User-defined	Specifies the web service method you want to test.
Method Endpoint	User-defined	Specifies the endpoint that the selected method belongs to.

Property	Value	Description
Inputs	User-defined	Specifies the input parameters configured for the selected method.
Outputs	N/A	Specifies the results of the method test.
Test	System-defined	Click to execute the selected method to ensure that it behaves as expected.

Web Service Refresh

The **Refresh** tab is used to change the current web service definition by using either a different WSDL file or an updated version of the same file. If there is no change, the web service is not refreshed.

The web service must be refreshed if one of the following has occurred:

- The web service has been moved to a different server.
- The server is renamed.
- The web service is hosted at a different URL.
- The web service metadata has changed (for example, the Service Contract, Operation Contract, or the Data Contracts).



Important:

After the web service is refreshed you must restart the Workflow application server and any clients in order to apply the changes.

The following table describes the information available for refreshing a web service.

Property	Value	Description
WSDL URL	User-defined	Specifies the address of the WSDL file associated with the web service.
Refresh	System-defined	Click to update the WSDL file associated with the current web service definition.

Limitations for Web Services Service Provider

Operators may experience one or more issues when using the Web Services Service Provider (WSSP). This topic provides information to workflow authors about the issues that operators may experience during run time due to design-time configurations.

Usage Limitations

Consider the following run-time issues when configuring WSSP.

Issue	Description
Inability to download secure WSDLs	Operators without the appropriate security permissions cannot download WSDL metadata from security-enabled web services.
Inability to duplicate web services connections	Operators cannot duplicate a connection because this option is not enabled.
Inability to edit the web services name	Operators cannot modify the name of the existing web service connection because this option is not enabled.
Inability to edit the web services configuration file	Operators cannot make permanent changes to the WSSP connection during run time because they do not have access to the related configuration file.
Inability to restore a web services database	Operators cannot restore a database with web services proxies because those proxies are not saved on disk. The web service must be deleted and created again.

Legacy Workflow Web Service

Legacy Workflow Web Service

Workflow publishes a web service, `IWorkflowWebService`, that allows schedule operations. You can browse all configured workflow schedules and start one or more of them, as well as get information on running workflow instances.

Endpoints are published using the following address:

```
https://<ServerName>:8202/SOAServer/SOAPProject/Workflow/Workflow/
IWorkflowWebService
```

The service metadata (WSDL) can be accessed at the following address:

```
https://<ServerName>:8202/SOAServer/SOAPProject/Workflow/Workflow/IBrowse?wsdl
```

If you are working in a *multiple server* environment, ensure that the web server is on the workflow server:

```
https://<ServerName>:8202/WorkflowServer/SOAPProject/Workflow/Workflow/IBrowse?
wsdl
```



Note:

Administrators are unable to log in using the web service interface.

Methods

These operations are provided by the Workflow web service. You can also use the related data contracts to customize these methods when developing workflows.



Important:

The Call Web Method that was provided in earlier versions of Workflow is still available. You can continue to use it for your legacy web services; however, it is recommended that for all new web services configurations, you should use the Web Services Service Provider to add and configure web service definitions.

GetFilteredWorkflowSchedules

Filters a set of schedules and returns only those schedules that match a regex filtering expression. It returns a collection of data contract objects from WorkflowScheduleData.

Inputs	Data Type	Description
scheduleFilter	String	Specifies the regex filter to filter the schedules by.

Outputs	Description
Collection<WorkflowScheduleData>	Returns a list of preconfigured workflow schedules that match a regex filtering expression.

Data Contract

Properties of the WorkflowScheduleData object. This object is returned by the GetFilteredWorkflowSchedules method.

```
[DataContract]
public class WorkflowScheduleData
{
    [DataMember]
    public string DisplayName;

    [DataMember]
    public string UserVersion;

    [DataMember]
    public string ScheduleAddress;
}
```

GetFilteredWorkflowSchedulesEx

Returns a list of workflow schedules.

Inputs	Data Type	Description
ScheduleFilter	ScheduleFilter	Specifies the criteria by which to filter the schedules.

Outputs	Description
Collection<WorkflowScheduleData>	Returns a list of preconfigured workflow schedules.

Data Contract

The ScheduleFilter object passes into the GetFilteredWorkflowSchedulesEx method and provides the required parameters to this method.

```
[DataContract]
public class ScheduleFilter
{
    [ DataMember ]
```

```

public string  DisplayNameFilter;

[ DataMember ]

public bool  UseStartTaskFilter;

}

```

StartWorkflowScheduleWithResult

Starts the named workflow schedule with specified input parameter values, and returns a workflow instance ID. The client can then make additional requests against this instance at a later time.

Inputs	Data Type	Description
startParams	StartScheduleParameters	Specifies the parameters of the schedule being started.

Data Type	Description
GUID	Returns the workflow instance ID of the instance that was started.

Data Contracts

The StartScheduleParameters object passes into the StartWorkflowScheduleWithResult method and provides the required parameters to this method. It contains the following properties and uses the WorkflowIdentifier ENUM.

```

[DataContract]

public class StartScheduleParameters
{
    [DataMember]

    public WorkflowIdentifier ScheduleId;

    [DataMember]

    public Collection<ScheduleInputParameterOverride> InputParameterOverrides;
}

```

Schedules start based on the WorkflowIdentifier values requested. This data contract can contain either the address of the schedule or its display name and user version. The type of data being sent is specified by the Type parameter of WorkflowIdentifier. If this parameter is not set and no address is provided, then the schedule start defaults to Address and throws an exception.

```

[DataContract]
public class WorkflowIdentifier
{
    [DataMember]
    public string Address;

    [DataMember]
    public string DisplayName;

    [DataMember]
    public string UserVersion;

    [DataMember]
    public WorkflowIdentifierType Type;
}

[DataContract]
public enum WorkflowIdentifierType
{
    [EnumMember]
    Address,
    [EnumMember]
    DisplayName
}

```

GetScheduleDetails

Allows you to search for a workflow schedule using either the schedule's address or its display name and user version. If a schedule is found, a list of all the schedule's parameters is returned.

Inputs	Data Type	Description
scheduleId	WorkflowIdentifier	Specifies the Address or Display Name and User Version of the workflow schedule you are retrieving details for. You must also provide the type of identifier (Address or Display Name).

Outputs	Description	
WorkflowScheduleDetails		Returns a list of schedule parameters, including the name, default value, and data type of each parameter.

GetWorkflowInstance

Accepts IDs for a workflow instance and a WorkflowInstanceData object containing the information relating to this workflow instance, and returns data related to the instance.

Inputs	Data Type	Description
instanceId	GUID	Accepts the ID for a given WorkflowInstance.

Outputs	Description	
WorkflowInstanceData		Returns data containing information on the status, start time (in descending order), and other related information on the instance. If the instance no longer exists in the database, it returns a null value.

Data Contracts

The WorkflowInstanceData object returns WorkflowInstance information, and contains the following properties.

```
[DataContract]
public class WorkflowInstanceData
{
    [DataMember]
    public WorkflowInstanceState State;

    [DataMember]
    public DateTime StartTime;
```

```

[DataMember]
public Guid Id;

[DataMember]
public string ScheduleAddress;
}

```

GetWorkflowInstances

Accepts IDs for a workflow instance and returns data related to the instance.

Inputs	Data Type	Description
schedule	WorkflowInstanceFilter	Accepts a given instance filter object and returns a collection of all workflow instances for that schedule.

Outputs	Description
Collection<WorkflowInstanceData>	Returns data containing information related to start time (in descending order). If the instance no longer exists in the database, it returns an empty collection.

Data Contracts

The WorkflowInstanceFilter object passes into the GetWorkflowInstances method and can filter by schedule address. You can impose limits on the number of results returned. It contains the following properties.

```

[DataContract]
public class WorkflowInstanceFilter
{
    [DataMember]
    public WorkflowIdentifier ScheduleId;

    [DataMember]

```

```
public int NumResults;
}
```

The WorkflowInstanceData object contains the following properties.

```
[DataContract]
public class WorkflowInstanceData
{
    [DataMember]
    public WorkflowInstanceState State;

    [DataMember]
    public DateTime StartTime;

    [DataMember]
    public Guid Id;

    [DataMember]
    public string ScheduleAddress;
}
```

Categories

Categories

Categories allow you to organize the resources associated with your workflows.



Note:

You assign categories directly from within a workflow. If a category has not already been created, you can also create it within the workflow you want to use it in. When you create a category from within a workflow, it is also added to the Categories folders list.

Resources that can be associated with workflows include:

- workflow
- schedules
- subprocesses
- user activities
- faults

- completion codes
- filters

When you assign categories to these resources, you can filter and search your project elements more effectively without having to browse for them. When you modify the name of a category resource, it is reflected in all associated resources.

Category Configuration

Category configuration provides general information about the currently selected category.

Attributes

Using *Associated Resources*, you can display workflow resources that have a category assigned to them.

Assign a new category resource to a workflow

Before you begin



Note:

- Categories can also be added directly to the Categories folder.
- Category names should not contain spaces.
- You can add as many category resources as required.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. On the **Overview** tab, click **Add/Remove Categories**.
6. Click **Add New Category**.
The Add Category dialog box appears.
7. In the **Name** and **Description** fields, enter a name and description for the resource.
8. Click **OK**, and then click **Save**.

Results

The new category appears in the Selected Categories box, as well as in the Categories folder of the navigator.

Assign an existing category resource to a workflow

Before you begin



Note:

You can add as many category resources as required.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. On the **Overview** tab, click **Add/Remove Categories**.
6. In the **Available Categories** list, select the category resource you want to add to the workflow, and then click the right arrow key.
7. Click **Save**.

Remove a category resource from a workflow

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. On the **Overview** tab, click **Add/Remove Categories**.
6. In the **Selected Categories** list, select the category resource you want to remove from the workflow, and then click the left arrow key.
7. Click **Save**.

Completion Codes

Completion Codes

Completion codes indicate how a subprocess finishes, and may change depending on the outcome of the subprocess execution.

Any information the completion code provides is pre-set and user-defined. This information is referred to as values. You can define a set of completion codes for workflows in a project. Each subprocess has a built-in completion code property that can be preconfigured, depending on your needs. In general, if a subprocess completes as X, then set the completion code value to state X. If a subprocess does not complete as X, then set the completion code to state Y.

Example

If the default value of the completion code property is set to "Success," and the subprocess finishes as expected, the completion code value remains as "Success." If the default value of the completion code property is set to "Success," and the subprocess encounters a problem during execution, the pre-set completion code value becomes "Failure." You would have preconfigured the values "Success" and "Failure" in the workflow development stage to provide more information about the outcome of the subprocess.

Completion codes are also important for reporting purposes. They store the subprocess history, or the result of the subprocess. This information can be included in history reports.

Completion codes can also be used to make decisions in other parts of a workflow's process. That is, you can compare a subprocess' completion code property to one of the defined completion codes in an If/Else condition to decide which workflow path to follow.

Related information

[Task Step \(Subprocess\) Configuration \(on page 1304\)](#)

Completion Code Uses

Completion codes tell you all the possible states that a workflow can end in.

These codes are user-defined. Completion codes are defined by two types of values: a default value and a value. The default completion code value is the value given to the *CompletionCode* property of the subprocess before the subprocess is executed. The value for a completion code property indicates the end result of your subprocess. If you do not set the default value of a subprocess' completion code or

write the value using an activity, the value of the completion code property is not affected. The default setting remains undefined.

You create completion codes in the navigator to use in workflows. Examples of possible completion codes are:

- Completed without faults
- Completed with faults
- Completed by user X
- Success
- Failure

You can view values that have been assigned to a subprocess during a debugging session. As you step through a workflow, the Variable Viewer in *Debug View* displays this information.

Common Uses

Completion codes are resources that make up a standard set of values. These values are used with completion code subprocess properties to indicate how a subprocess has finished. You can set one of the values that has been preconfigured in the standard set of values to the completion code property. This is done through the *Write activity*.

You can also test and/or compare the value of the completion code property to a standard set of values using *Workflow expressions*. In general, any activity that allows you to create an expression allows you to use a completion code. The most common place where a completion code is used within a subprocess is in an *If/Else activity*. Because If/Else allows for conditional resolutions, you can use this activity to check the values of a specified completion code, as well as to check to see how another activity within a subprocess completed. To do this, use the equality or inequality operator in an expression to compare the value of the completion code property to one of the values in the completion code set.

Properties and Subprocesses

There is a completion code property on every local or global subprocess of a workflow. There can be only one completion code for each subprocess.

Related reference

[Write Activity \(on page 1447\)](#)

[Workflow Expressions \(on page 1286\)](#)

[If/Else Activity \(on page 1486\)](#)

Related information

[Workflow Debug View \(on page 1567\)](#)

Set a default value for a subprocess' completion code property

Before you begin**Note:**

Completion codes must already have been added to the Completion Codes folder.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. To access activities, double-click the subprocess you want to work with.
6. On the subprocess activity configuration panel, in the **Default Completion Code** field, click the ellipsis button.
The Edit Completion Code dialog box appears.

**Note:**

You can use only one completion code resource per subprocess.

7. Select a completion code value, and then click **OK**.
8. Click **Save**.

Set a value for a subprocess completion code property

Before you begin**Note:**

Completion codes must already have been added to the Completion Codes folder.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.

4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. To access activities, double-click the subprocess you want to work with.
6. Select or insert a *Write activity*.
7. Click the ellipsis button.
The **Select Variable** dialog box appears.
8. Click the **Subprocess** tab, and then double-click **Properties**, and then click **Completion Code**.
9. On the Write activity configuration panel, in the **Value** field, click the ellipsis button.
The Edit Completion Code dialog box appears.

**Note:**

You can use only one completion code resource per subprocess.

10. Select the value for the completion code property, and then click **OK**.
11. If required, to bind the value, click the arrow and select **Bind**.
 - a. In the **Select Variable** dialog box, click the **Subprocess** or **Workflow** tab.
 - b. Select the completion code you want to bind to.
 - c. Click **OK**.
12. Click **Save**.

Related reference

[Write Activity \(on page 1447\)](#)

Use completion codes in an expression

Before you begin**Note:**

Completion codes must already have been added to the Completion Codes folder.

Procedure

1. In the navigator, click **Workflow**.
2. In the **Folders** list, select a resource type, such as Workflows, Subprocesses, Schedules, User Activities, Categories, Completion Codes, Faults, or Filters.
3. In the resource type list, select a resource.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. To access activities, double-click the step you want to work with.
6. Select or insert an *If/Else* or *While* activity.
7. On the activity configuration panel, click **Edit Condition**.

The Expression Editor appears.

8. Click the **Workflow Variables** tab, and then click the **Subprocess** tab.

9. To create an expression in the **Enter Expression** field:

a. From the **Properties** list, double-click **Completion Code**.

b. Insert an operator.



Note:

You can only use the equality (==) and inequality (!=) operators with completion code properties and values.

c. Click the **Completion Codes** tab, and select a completion code to complete your expression.

10. To verify your expression:

a. Click **Check Syntax**.

b. Click **Test**.

c. Click **Evaluate**.

d. Click **Close**.

e. Click **OK**.

11. click **Save**.

Related reference

[If/Else Activity \(on page 1486\)](#)

[While Activity \(on page 1480\)](#)

Actuals

Actuals

Resource actuals show the as-performed actual values for a resource during a production cycle. A resource actual is associated with a resource specification, making it easy to compare actual and specified values for properties.

When the planned use of resources does not match the actual use of resources, then an actual is used. The actual records the true values of resources during production, which can be compared to expected use. Work process segments have specifications, but no way to record actual values. Using actuals makes it possible to take a true measurement of the resources that were used.

Each actual corresponds to an existing equipment, material, or personnel resource.

IProductionRuntime Service Interface

Creating actuals and editing their properties is managed using call methods of the IProductionRuntime Service Interface. For more information about these properties, see **Production Runtime Call Methods**.

Example

Suppose you have a workflow for attaching a wheel to a hub. You specify in the work process definition that during the production process, one wheel/rim assembly and six lug nuts are consumed. Now, you want to write an application that tracks your inventory of lug nuts, so you can replace them when needed. One way to do that would be to simply subtract six lug nuts from inventory every time the process runs. But sometimes a lug nut is defective, or was stripped while the operator tried to add it, and another one is needed. Now seven lug nuts have been consumed. With an actual, you can record that seven lug nuts were used instead of six, so the inventory count can be maintained accurately.

By having these values associated with the specifications, you can also tell when your production is in control. If your inventory seems to be shrinking faster than it should, you can write queries to return the difference between actual and specified values for the last month's work requests. You can use the queries to spot the point where you are using more resources than you specified. You can then drill down through the work responses looking for correlations with other specifications. Maybe you find that the spoilage rate for lug nuts goes up whenever an operator used air hammer #2. So you order an inspection of the hammer, and find out that it's not sealing the lug nuts correctly due to a flaw, and fix it.

Specification properties and their actual properties go a level deeper. Suppose you specify that a beer batch must cook for 2 hours at 135 degrees. Then your final beer batch goes through quality assurance and does not pass inspection. How can you determine what went wrong? You can run a query that compares actual measured values against specification values, to find anomalies. Your query shows that the actual measured temperature during cooking was 110 degrees, not the specified 135 degrees. So you inspect the cooker, and find out it has a defective thermostat.

Create a resource actual

This procedure describes all of the steps required to add and configure resource actuals for a specified segment response.

Before you begin

Depending on the type of actual you are creating, before you proceed with this procedure, make sure you have created the appropriate equipment, material, or personnel resource you are configuring the actual(s) for.

About this task

You can create expressions or bind properties for many of these properties, depending on your workflow requirements.



Attention:

The following procedure is provided as an example only and is not meant to be used in a production environment.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. Click **Add**, and then add and configure a workflow as follows:
 - a. On the workflow configuration panel, in the **Personnel** field, click the ellipsis button, and then select the personnel resource you want to assign to this workflow.
 - b. Click **Edit** next to **Parameters**.
The **Edit Parameters** dialog box appears.
 - c. Click **Add**, and then configure an input parameter with a **Type** of `String`, a **Direction** of `In`, and a unique name to bind to the **S95 Id** of the work request you will create in step 5b.
 - d. Click **Add**, and then configure an output parameter to represent the resource actual's display name, with a **Type** of `String`, a **Direction** of `Out`, and a unique name to bind to the Write activity you will add in step 20d.
 - e. Click **Add**, and then configure an output parameter to represent the results of the workflow, with a **Type** of `Boolean`, a **Direction** of `Out`, and a unique name to bind to the Write activity you will add in step 20h.
 - f. Click **OK**.
 - g. Add a local subprocess to the workflow, click **Edit** on the subprocess configuration panel, enter a unique name to represent the resource actual configuration, and then click **OK**.
 - h. Add another local subprocess after the first, click **Edit** on the subprocess configuration panel, enter a unique name to represent the query you will configure in step 9, and then click **OK**.
3. Double-click the resource actual subprocess, and then insert a Call Method activity.
4. On the activity configuration panel, click **Configure Method**.

The **Configure Method** dialog box appears, displaying a list of the available call method service providers.

5. In the **IProductionRuntime** methods, click **CreateWorkRequest**, click **Finish**, and then configure the input parameters.
 - a. **Optional:** In the **Description** field, enter information to uniquely describe the work request.
 - b. In the **S95 Id** field, bind to the input parameter you configured for the workflow in step 2c.
 - c. In the **Work Definition** field, click the ellipsis button, and then select the work definition to associate with the work request.
6. Insert another Call Method activity, and then click **Configure Method**.
7. In the **IProductionRuntime** methods, click **CreateWorkResponse**, and then click **Next**.
 - a. In the **Item specific configuration** area, select a work type from the drop-down list, and then click **Finish**.
 - b. On the activity configuration panel, click **Edit**, enter a unique name to represent the call method, and then click **OK**.
 - c. **Optional:** In the **Description** field, enter information to uniquely describe the work response.
 - d. In the **Location** field, click the ellipsis button, and then select the equipment location to associate with the work response.
 - e. In the **S95 Id** field, enter a name for the work response, or click the arrow and create an expression; for example, "`<workresponsename>`" + `Tostring(CurrentDateTime())`, which will create a unique name for every work response that is created when this workflow is run. The value of "`<workresponsename>`" is the name you choose to give to the work responses.
 - f. In the **Work Request** field, click the arrow, and then bind the work response to the work request you created in step 5.
8. Insert another Call Method activity, and then click **Configure Method**.
9. In the **IProductionRuntime** methods, select **CreateSegmentResponse**, and then click **Finish**.
 - a. On the activity configuration panel, click **Edit**, enter a unique name to represent the call method, and then click **OK**.
 - b. **Optional:** In the **Description** field, enter information to uniquely describe the segment response.
 - c. **Optional:** In the **End Time** field, click the arrow, and then select the terminating date and time for the segment response.

- d. In the **Location** field, click the ellipsis button, and then select the equipment location to associate with the segment response.
 - e. In the **Parent Id** field, click the arrow, and then bind the segment response to the output parameter of the work response you created in step 6.
 - f. In the **S95 Id** field, enter a name for the segment response, or click the arrow and create an expression; for example, "<segmentresponsename>" + ToString(CurrentDateTime()), which will create a unique name for every segment response that is created when this workflow is run.
The value of "<workresponsename>" is the name you choose to give to the segment responses.
 - g. **Optional:** In the **Start Time** field, click the arrow, and then select the starting date and time for the segment response.
 - h. In the **Work Process Segment** field, click the ellipsis button, and then select the work process segment to associate with the segment response.
10. Insert another Call Method activity, and then click **Configure Method**.
11. In the **IProductionRuntime** methods, select the method applicable to the type of actual you are creating, based on your need to set initial start and end time values:

If you...	Then select one of the following
Need to set initial start and end time values	<ul style="list-style-type: none"> ◦ CreateEquipmentActualEx ◦ CreateMaterialActualEx ◦ CreatePersonnelActualEx
Do not need to set initial start and end time values	<ul style="list-style-type: none"> ◦ CreateEquipmentActual ◦ CreateMaterialActual ◦ CreatePersonnelActual

12. Click **Finish**.
13. Configure the call method as follows:
- a. On the activity configuration panel, click **Edit**, enter a unique name to represent the call method, and then click **OK**.
 - b. **Optional:** In the **Description** field, enter information to uniquely describe the resource actual you want to create.

- c. If the call method you are configuring allows you to designate the initial **End Time** value, click the drop-down arrow to set the date and time.
- d. Depending on the type of resource actual you are creating, in the **Equipment**, **Material**, or **Personnel** field, click the ellipsis button, select the appropriate resource for the actual, and then click **OK**.
- e. **Optional:** If you are creating a material resource actual, then, in the **Lot or Sublot ID** field, enter a literal value, or click the drop-down arrow to perform a bind or create an expression.



Note:

The data type of this field must be GUID.

- f. In the **Quantity** field, enter a literal value, or click the drop-down arrow to perform a bind or create an expression.
- g. **Optional:** In the **Requirement ID** field, enter the resource requirement you want to associate with the actual, or click the drop-down arrow to perform a bind or create an expression.



Note:

The data type of this field must be GUID.

- h. In the **S95 ID** field, enter the unique name of the resource actual as you want it to appear on the segment response.
 - i. If the call method you are configuring allows you to designate the initial **Start Time** value, click the drop-down arrow to set the date and time.
 - j. In the **Segment Response ID** field, click the drop-down arrow and bind the value to the **ReturnValue** of the output parameter of the segment response you created in step 9.
 - k. In the **Unit of Measure** field, enter the unit of measure you want to associate with the resource actual.
14. Click **Save**.
15. Return to the workflow level, and then double-click the subprocess you added in step 2h, and then configure it as follows:
- a. On the subprocess configuration panel, click **Edit** next to **Local Variables**.
The **Edit Local Variables** dialog box appears.

- b. Click **Add**, and then configure a variable with a **Type** of `String`, an **Initial Value** of `**`, and a unique name to bind to the Write activity you will create in step 20e.
 - c. Click **OK**.
16. Insert a Call Method activity.
17. On the activity configuration panel, click **Configure Method**.

The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
18. In the **IProductionRuntime** methods, click **QuerySegmentResponses**, click **Finish**, and then configure the input parameters.
 - a. On the activity configuration panel, click **Edit**, enter a unique name to represent the call method, and then click **OK**.
 - b. From the **Detail Level** drop-down list, select `FullDetails`.
19. Insert a For Each activity after the QuerySegmentResponses Call Method activity, and then configure it as follows:
 - a. On the activity configuration panel, click **Edit**, enter a unique name to represent the activity, and then click **OK**.
 - b. In the **Collection** field, click the arrow, bind to the return value of the call method you added in step 9, and then click **OK**.
20. Insert another For Each activity after the first For Each activity, and then configure it as follows:
 - a. On the activity configuration panel, click **Edit**, enter a unique name to represent the activity, and then click **OK**.
 - b. In the **Collection** field, click the arrow, bind to the resource actual property of the For Each activity you added in step 19, and then click **OK**.

Be sure to select the appropriate property for the type of resource actual you are creating. For example, if you are creating an equipment actual, select the EquipmentActuals property.
 - c. Insert a Write activity inside the current For Each activity, and then add two writes.
 - d. Bind the first write to the workflow output parameter you configured in step 2d, bind the value of the write to the **DisplayName** property of the current For Each activity, and then click **OK**.
 - e. Bind the second write to the DisplayName property of the local variable you created in step 15b, bind the value of the write to the **DisplayName** property of the current For Each activity, and then click **OK**.

- f. Insert an If/Else activity after the Write activity.
 - g. For the left branch of the If/Else activity, click **Edit** on the activity configuration panel to enter a unique name for the condition, and then click **Edit Condition** to enter an expression that represents the result the workflow should evaluate to; for example, `{Query Segment Responses}.{Equipment Display Name} == "01 - Motherboard Attach"`.
 - h. Insert a Write activity within the left branch of the If/Else activity after the condition, click **Edit** on the activity configuration activity panel to enter a unique name for the write activity, bind the write to the output parameter you created in step 2e, and then from the **Value** drop-down list, select `True`.
21. Click **Save**.
 22. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.
 23. You can view the resource actuals by navigating to the work request you created with this workflow and clicking the **Response Information** tab in the **S95 Model Editor**. For more information, see *Query segment responses*.

Add resource actual properties

Before you begin

After you create a workflow, you can add and configure actuals using call methods for service providers. You can create expressions or bind properties for many of these properties, depending on your actual requirements.



Note:

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with



Note:

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.

5. Insert an activity:

◦ At the workflow level:

- From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

◦ At the subprocess level:

- Double-click the local subprocess you want to add the activity to.
- From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

6. On the activity configuration panel, click **Configure Method**.

The **Configure Method** dialog box appears, displaying a list of the available call method service providers.

7. Double-click the service provider you want to use, and then select the call method.

8. Click **Next**.

9. In the **Item specific configuration** section, configure the method.



Note:

Item configuration depends on the selected source and item. Some items do not require configuration.

10. At run time, if you want to return all data from the selected item, select the **Return all data** option.

11. At run time, if you want to return a subset of data from the selected item at run time, select the **Return only a subset of data** option, and then from the data tree, select the data you want returned.



Note:

If the item returns only a single piece of data, the **Return all data** option is selected, and you cannot retrieve a subset of data.

12. Click **Finish**.

13. If required, on the activity configuration panel, modify the inputs and/or create an expression or perform a bind for the selected call method.

14. In the IProductionRuntime methods, select **AddResourceActualProperties**.

15. Click **Finish**.

16. In the Actual ID field, enter a literal value, or click the drop-down arrow to perform a bind.

**Note:**

The data type of this field must be GUID.

17. In the Property Names field, perform one of the following:

- a. Click the ellipsis button.
- b. In the String Collection Editor dialog box, enter a series of strings for the names of the resource properties to add, and then click **OK**.

or

- Click the drop-down arrow to perform a bind, and then click **OK**.

**Note:**

The data type of this field must be String[].

18. In the Resource Type field, select a resource type from the drop-down menu.

19. Click **Save**.

REST APIs

REST APIs

You can use the REST APIs to interact with the Workflow application over the Internet.

REST APIs for User Activities

Execute a User Activity

Executes the identified user activity.

HTTP Method	URL	Parameters	Request Body	Response Body
POST	https://server-Name:port/Proficiency-WebApi/api/User-Activities/<id> ?action=execute&timeout=30	<p>Values are in JSON format.</p> <p>Action: <code>execute</code> is the only supported value.</p> <p>Timeout: amount of time to wait for</p>	<p>Input values.</p> <pre>{ "Input1" : 1, "Input2" : "hello" }</pre>	<p>Output values.</p> <pre>{ "Output": 1 }</pre>

HTTP Method	URL	Parameters	Request Body	Response Body
		results in seconds. To return immediately without a result, use 0, or do not specify.		

Get All User Activities

Returns the names and IDs of all user activities.

HTTP Method	URL	Default Port	Return Value
GET	https://server-Name:port/ProficyWebApi/api/UserActivities	8444	Returns the name and ID of all user activities defined in the system, in JSON format. Each item includes a link to its metadata.

```
{
  "Id": "00b8dc5c-ed03-4654-ab64-b841e72bf681",
  "DisplayName": "Do Other Stuff",
  "Links": [{
    "Rel": "self",
    "Href": "https://localhost:8444/ProficyWebApi/api/UserActivities/00b8dc5c-ed03-4654-ab64-b841e72bf681"
  }]
}, {
  "Id": "a5857385-51e5-442b-8764-84d4080b9bda",
  "DisplayName": "Do Stuff",
  "Links": [{
    "Rel": "self",
    "Href": "https://localhost:8444/ProficyWebApi/api/UserActivities/a5857385-51e5-442b-8764-84d4080b9bda"
  }]
},
}
```

Get Metadata for a Single User Activity

Returns the defined metadata for a single user activity.

HTTP Method	URL	Default Port	Return Value
GET	https://server-Name:port/ProficyWebApi/api/UserActivities/<id>	8444	Returns the following user activity metadata: name, ID, input parameters (name, type), and output parameters (name, type).

```
{
  "InputParameters" : [{
    "Name" : "Parameter1" ,
    "Type" : "System.Int32 "
  }],
  "OutputParameters" : [{
    "Name" : "Parameter2" ,
    "Type" : "System.Int32 "
  }],
  "Id" : "a5857385-51e5-442b-8764-84d4080b9bda" ,
  "DisplayName" : "Do Stuff",
  "Links": [{
    "Rel": "self",
    "Href": "https://localhost:8444/ProficyWebApi/api/UserActivities/a5857385-51e5-442b-8764-84d4080b9bda"
  }], {
    "Rel": "UserActivities",
    "Href": "https://localhost:8444/ProficyWebApi/api/UserActivities"
  }
  ]
}
```

User Activity Error Codes

If a REST operation fails, additional information about the error is available from both the response Error-Code header and the response body.

Code	Description	
	1020	Specifies that an unsupported value was provided for the action parameter.
	1021	Specifies that a parameter value was not supplied.
	1022	Specifies that supplied parameter value was an incorrect type.
	1023	Specifies that the execution of the user activity timed out before it completed.
	1024	Specifies that user activity terminated during execution.

REST API for E-signatures

Create a New E-signature Record

You can create a signature record based on a provided definition. A record must be created before it can be signed.

HTTP Method	URI Example	Default Port	Request Body	Return Value
PUT	https://server-Name:port/ProficiencyWebApi/api/signatures	8444	Definition provided as JSON data. <pre> { "ValidPerformers": "PerformerGroupName", "VerifierRequired": "True", "ValidVerifiers": "VerifierGroupName", "Description": "Signing required" } </pre>	Returns the signature record ID.

Sign an E-signature Record

The signature record can be signed once for the performer, and, if required, a second time for the verifier. When all signatures are complete, the record is closed. To sign an existing signature record, you use the signature ID that is returned from the creation request.

HTTP Method	URI Example	Default Port	Request Body
POST	https://server-Name:port/ProficyWebApi/api/signatures/<id>	8444	Signing information provided as JSON data. <pre>{ "UserName": "UserName", "Password": "UserPassword", "Comment": "I am signing this" }</pre>

Get an E-Signature Record

You can retrieve an in-progress or completed signature record.

HTTP Method	URI Example	Default Port	Return Value	Status Parameter
GET	https://server-Name:port/ProficyWebApi/api/signatures/<id>	8444	Signature information provided as JSON data. <pre>{ "Id": "fff91ac0-6df0-4f 40-b7eb-1526ec13e5c 7", "Status": 2, "PerformerSignature": { "UserName": "user",</pre>	0: Awaiting Performer 1: Awaiting Verifier 2: Completed 3: Cancelled 4: Failed

HTTP Method	URI Example	Default Port	Return Value	Status Parameter
			<pre> "SigningTime": "2016-05-26T18:30:3 2.017", "Comment": "I am signing this" }, "VerifierSignatur e": { "UserName": "admin", "SigningTime": "2016-05-26T18:30:4 8.03", "Comment": "I am verifying this" } } </pre>	

Cancel an E-Signature Record

You can cancel an in-progress signature record.

To cancel an existing signature record, use the signature ID that is returned from the creation request. A cancel action parameter, **cancelSignature**, is required and its value should be `true`; otherwise the signature record won't be cancelled.

HTTP Method	URI Example	Default Port	Status Parameter
POST	<a href="https://server-Name:port/ProficyWebApi/api/signatures/?id=<signatureid>&cancelSignature=true">https://server-Name:port/ProficyWebApi/api/signatures/?id=<signatureid>&cancelSignature=true	8444	0: Awaiting Performer 1: Awaiting Verifier

E-Signature Error Codes

If a REST operation fails, additional information about the error is available from both the response Error-Code header and the response body.

Signing Errors

Code	Description	
	1000	Specifies that the resource ID that has been provided is not in a valid format.
	1001	Specifies that the credentials provided for the signing are invalid.
	1002	Specifies that the signer is not a member of the required group to perform the signing.
	1003	Specifies that the user has already signed as the performer and is attempting to sign as the verifier. The same user cannot both sign and verify the same record.
	1004	Specifies that the signer has been disallowed by the electronic signature system.
	1005	Specifies that all signings have already been completed for the related signature record.
	1006	Specifies that the signing information provided in the request body is malformed.
	1007	Specifies that an unknown error has occurred during signing.

Record Creation Errors

Code	Description	
	1011	Specifies that an invalid group name has been designated as the Performer Group.
	1012	Specifies that an invalid group name has been designated as the Verifier Group.
	1013	Specifies that no description has been provided for the signature.
	1014	Specifies that a task, task step, or form ID was not included when the signature was created, and that the definition was not provided in the body.
	1015	Specifies that a signature has not been configured for task, task step, or form indicated.

REST API for Equipment

All equipment methods use the same base URL.

Base URL

https://<serverName>:<port>/ProficyWebApi/api/Equipment

- <serverName> is the hostname where the Proficy SOA server is running.
- <port> default is 8444.

Authentication is required on each request. Basic authentication with username and password must be provided in the Authorization header.

Get Equipment Instances

Returns a list of equipment instances that matches the provided criteria. The hierarchy is also returned to allow for duplicate names for equipment instances.

Request Parameters

At least one of the following parameters must be specified in the request.

Parameter	Type	Required	Description
equipmentName	String	Y/N	Retrieve equipment instances with this display name.
className	String	Y/N	Retrieve equipment instances that are associated with this equipment class.

The following example displays all equipment instances with the specified display name.

HTTP Method	GET
URL	https://serverName:port/ProficyWebApi/api/Equipment?equipmentName=<display name>
Return Value	Returns all equipment instances with the specified equipment name.
Sample Response	<pre>{ { "Id": "0cb0632b-04c1-4f16-9620-44aaa6e0d0a0", "Name": "Areal", "Description": "", "Hierarchy": "Enterprise/Site/Areal", "Classes": ["Class1", "Class2"], "Links": [{ "Rel": "self", "Href": "https://sql2014server:8444/ProficyWebApi/api/Equipment/0cb0632b-04c1-4f16-9620-44aaa6e0d0a0" }] } }</pre>

```

    }
  ]
},
{
  "Id": "fcbae6fd-36df-4997-88e5-4a39eddc760c",
  "Name": "Areal",
  "Description": "in Enterprise2",
  "Hierarchy": "Enterprise2/Site/Areal",
  "Classes": [],
  "Links": [
    {
      "Rel": "self",
      "Href":
        "https://sql2014server:8444/ProficyWebApi/api/Equipme
        nt/fcbae6fd-36df-4997-88e5-4a39eddc760c"
    }
  ]
}
}
}
}

```

The following example displays all equipment instances with the specified display name and that are associated with the specified class name.

HTTP Method	GET
URL	https://serverName:port/ProficyWebApi/api/Equipment?equipmentName=<display name>&className=<equipment class>
Return Value	Returns all equipment instances with the specified equipment name and class name.
Sample Response	<pre> [{ "Id": "0cb0632b-04c1-4f16-9620-44aaa6e0d0a0", "Name": "Areal", "Description": "", "Hierarchy": "Enterprise/Site/Areal", "Classes": [</pre>

```

    "Class1",
    "Class2"
  ],
  "Links": [
    {
      "Rel": "self",
      "Href":
        "https://sql2014server:8444/ProficyWebApi/api/Equipment
        /0cb0632b-04c1-4f16-9620-44aaa6e0d0a0"
    }
  ]
}
]

```

Get Single Equipment Instance

Returns the details of a specific equipment instance and, optionally, the properties. The hierarchy is also returned to allow for duplicate names for equipment instances.

Request Parameters

The following parameter is optional in the request.

Parameter	Type	Required	Description
returnProperties	Boolean	N	Indicates whether to include equipment property details in the response. Valid values are <code>true</code> or <code>false</code> . The default is <code>true</code> .



Note:

Where `<id>` is the unique identifier for the equipment instance.

The following example displays the details for the specified equipment instance, including its properties.

HTTP Method	GET
URL	<code>https://serverName:port/ProficyWebApi/api/Equipment/<id></code>
Return Value	Returns the details for the specified equipment instance, including the properties.

**Note:**

The **returnProperties** parameter is not required when you want to include equipment instance properties in the request.

Sample Response

```
{
  "Properties": {
    "Class1-P1": {
      "Description": "",
      "UnitOfMeasure": "",
      "Value": "p1Value",
      "Links": [
        {
          "Rel": "self",
          "Href":
            "https://sql2014server:8444/ProficyWebApi/api/Equipment
            /0cb0632b-04c1-4f16-9620-44aaa6e0d0a0?propertyName=Class
            s1-P1"
        }
      ]
    },
    "ColorProperty": {
      "Description": "Color changed to green",
      "UnitOfMeasure": "",
      "Value": "Green",
      "Links": [
        {
          "Rel": "self",
          "Href":
            "https://sql2014server:8444/ProficyWebApi/api/Equipment
            /0cb0632b-04c1-4f16-9620-44aaa6e0d0a0?propertyName=Color
            Property"
        }
      ]
    },
    "PropertyInt32": {
      "Description": "",
```

```

        "UnitOfMeasure": "",
        "Value": 46,
        "Links": [
            {
                "Rel": "self",
                "Href":
                "https://sql2014server:8444/ProficyWebApi/api/Equipment
                /0cb0632b-04c1-4f16-9620-44aaa6e0d0a0?propertyName=Prope
                rtyInt32"
            }
        ]
    },
    "Id": "0cb0632b-04c1-4f16-9620-44aaa6e0d0a0",
    "Name": "Areal",
    "Description": "",
    "Hierarchy": "Enterprise/Site/Areal",
    "Classes": [
        "Class1",
        "Class2"
    ],
    "Links": [
        {
            "Rel": "self",
            "Href":
            "https://sql2014server:8444/ProficyWebApi/api/Equipment
            /0cb0632b-04c1-4f16-9620-44aaa6e0d0a0"
        }
    ]
}

```

The following example displays the details for the specified equipment instance, but does not include its properties.

HTTP Method	GET
URL	https://serverName:port/ProficyWebApi/api/Equipment/<id>?returnProperties=false

Return Value	Returns the details for the specified equipment instance, excluding the properties.
Sample Response	<pre> { "Properties": {}, "Id": "0cb0632b-04c1-4f16-9620-44aaa6e0d0a0", "Name": "Areal", "Description": "", "Hierarchy": "Enterprise/Site/Areal", "Classes": ["Class1", "Class2"], "Links": [{ "Rel": "self", "Href": "https://sql2014server:8444/ProficyWebApi/api/Equipment /0cb0632b-04c1-4f16-9620-44aaa6e0d0a0" }] } </pre>

Get Equipment Property by Name

Returns the details of the named property for the specified equipment instance.

Request Parameters

The following parameter must be specified in the request.

Parameter	Type	Required	Description
propertyName	String	Y	Retrieves the property with this name.

**Note:**

- Where <id> is the unique identifier for the equipment instance.
- Returns the details (Description, Unit of Measure, Value) of a single property.

HTTP Method	GET
URL	https://serverName:port/ProficyWebApi/api/Equipment/<id>?propertyName=<property name>
Return Value	Returns the details of the named property for the specified equipment instance.
Sample Response	<pre>{ "Description": "", "UnitOfMeasure": "", "Value": "Blue", "Links": [{ "Rel": "self", "Href": "https://sql2014server:8444/ProficyWebApi/api/Equipment /0cb0632b-04c1-4f16-9620-44aaa6e0d0a0?propertyName=Color Property" }, { "Rel": "parent", "Href": "https://sql2014server:8444/ProficyWebApi/api/Equipment /0cb0632b-04c1-4f16-9620-44aaa6e0d0a0" }] }</pre>

Update an Equipment Property

You can update the details of a property for a specific equipment instance.

Request Parameters

The following parameter must be specified in the request.

Parameter	Type	Required	Description
propertyName	String	Y	Update the details of the named property.

Where <id> is the unique identifier for the equipment instance.

HTTP Method	PUT
URL	https://serverName:port/ProficyWebApi/Equipment/<id>?propertyName=<property name>
Return Value	Updates the details of the named property for the specified equipment instance
Message Body	Property details are provided in JSON format in the message body. <pre>{ "Description": "Color changed to green", "UnitOfMeasure": "", "Value": "Green" }</pre>
Response Body	The response to a successful update is a status of 200 OK.

Equipment Error Codes

If a REST operation fails, additional information about the error is available from both the response Error-Code header and the response body.

Code	Description
1031	Specifies that equipment property name is empty.
1032	Specifies that no search criteria were provided.

Code	Description	
	1033	Specifies that equipment property details were not provided.
	1034	Specifies that the query parameter used is not known.

Chapter 11. eSOP

eSOP Orientation

eSOP Orientation

eSOP (Electronic Standard Operating Procedures) offers a simplified solution for managing and developing workflows. It increases the efficiency of your development process by using different skill levels to develop the final workflow product.

The simplified step-by-step process separates the complexity of workflow design from constructing and maintaining the actual task and task step content. An important advantage of eSOP is that users are able to be part of the basic workflow development process without knowledge of authoring the more complex structure of a workflow.

Two Types of Users

Workflow Designers

These users have in-depth understanding of the business and/or production processes in their organizations, as well as expertise in designing Workflow applications to support those processes. They are responsible for creating or adapting workflows for interface with the eSOP feature through specialized activities, and then designing templates that tie those workflows to step-by-step tasks developed in eSOP by the process experts.

Task Builders

These users have in-depth understanding of specific production processes in their organizations. They are responsible for constructing eSOP tasks that outline the steps required to complete the most commonly performed activities during production.

Related reference

[Design eSOP Workflows Overview \(on page 1647\)](#)

Related information

[Task Construction for Procedures \(on page 1664\)](#)

[Task Maintenance \(on page 1676\)](#)

Using this Documentation

This documentation is designed to familiarize you with the layout and use of the eSOP feature in Workflow.

There are two general types of eSOP users:

- *Workflow Designers*, who create or adapt workflows for interface with eSOP.
- *Task Builders*, who input the specific details to help a workflow designer create the final workflow used in production.

User Type	Description	Section or Topic
System Administrator	Outlines the approach taken in this documentation based on the type of user working with Workflow eSOP.	<i>Client Layout Configuraton for eSOP Users</i>
Workflow Designer and Task Builder	Describes the purpose of the eSOP feature, provides a high-level view of eSOP development, and outlines the user interface.	<i>eSOP Orientation</i>
Workflow Designer	Describes the specialized sub-processes and activities used in eSOP and provides detailed instructions for performing the activities required to configure templates for use in building eSOP tasks.	<i>Design eSOP Workflows Overview</i>
Task Builder	Provides detailed instructions for performing the activities required to configure step-by-step tasks in eSOP.	<i>Task Construction for Procedures</i>
Task Builder	Provides detailed instructions for performing the activities required to modify various aspects of eSOP tasks as your organizational processes change.	<i>Task Maintenance</i>

Related reference

[Design eSOP Workflows Overview \(on page 1647\)](#)

Related information

[Client Layout Configuration for eSOP Users \(on page 1631\)](#)

[eSOP Orientation \(on page 1625\)](#)

[Task Construction for Procedures \(on page 1664\)](#)

[Task Maintenance \(on page 1676\)](#)

eSOP Process Overview

This section provides an overview of the steps required to design and build workflows using eSOP.

About this task

The eSOP process has two levels of design. The Workflow Designer first must design and develop a workflow, including properties, parameters, activities, and so on. Then, the Workflow Designer must configure items in the eSOP model to provide an outline for the Task Builder to define implementations of the workflow in the form of tasks representing standard operating procedures.

Procedure

1. The Workflow Designer performs the following procedures.
 - a. Designs and develops a workflow, or uses the sample template that is part of eSOP.
 - b. Configures task templates that are used by the Task Builder to finalize the workflow configuration.
 - c. Designs custom configuration panels that are used by the Task Builder to finalize the workflow configuration.
 - d. Associates global subprocesses with task step templates.
2. The Task Builder performs the following procedures.
 - a. Translates established processes into step-by-step eSOP tasks based on templates authored by a Workflow Designer.
 - b. Establishes specific settings for the task, including associated personnel and equipment, expiration time, priority, and schedule.

**Note:**

Changing the template associated with a task may have implications for the configuration of the task, such as rendering your task invalid if there are more steps in the template than in the task.

- c. Delineates task-related correspondence in the form or work instructions, related documents, and initiation and/or expiration email messages.
- d. Specifies additional settings and/or information for the task, as required, based on unique organization needs and the custom configurations implemented in the underlying template to address those needs.
- e. Activates tasks for use as workflows at run time.

Related reference

[Design eSOP Workflows Overview \(on page 1647\)](#)

Related information

[Task Construction for Procedures \(on page 1664\)](#)

[Task Maintenance \(on page 1676\)](#)

Sample Files Available with eSOP Installation

Several sample files that demonstrate eSOP functionality are part of the eSOP feature. These files include a template workflow, comment forms, fault resources, and global subprocesses.

Recommendation

To use any sample resource, you must duplicate and re-name it before you modify it.



Important:

Samples may be updated in future versions to demonstrate new functionality and the upgrade process is expected to overwrite any modifications you have made to the samples.

When you modify an eSOP template workflow that has been associated with an eSOP task template, you must update the eSOP task template to ensure your task is compatible with the new workflow. You can do this by editing the eSOP task template, selecting the template workflow again, and saving it. This will update all tasks associated with the template.

Workflow Sample

A sample workflow template named `Sample Sequential Template` is part of installation. We recommend that you use this type of template as the basis of your eSOP workflow.

You can find the **Sample Sequential Template** in the navigator at **Workflow > Workflows**.

This template is configured to meet the requirements of most workflows. Workflow designers can use it as a foundation for building a workflow that will be used in eSOP. It demonstrates eSOP functionality such as comment forms, emails, keyword replacement, and jumps.

In a sequential template, the workflow subprocesses are all the same (so, no branching or looping is available at the workflow level). This template type allows you to configure jump options and move the order of the task steps configurations up or down using the eSOP Task Editor.

**Note:**

If there are more eSOP task steps than template subprocesses in this workflow, during run time, the workflow terminates.

Using this Template for Reporting

You can use this template to track workflow run-time data to meet your internal reporting needs. To use the workflow template for this purpose, copy the existing **Sample Sequential Template** workflow, rename it to a unique name, and then enable tracking on the **Tracking Configuration** tab. Then, in the workflow or subprocess list, select the variables you want to track.

Fault Samples

Three fault samples are part of eSOP. You can find these resources in the navigator at **Workflow > Faults**.

Exit Wait

Contained within a fault scope. If this type of fault occurs, the loop stops and skips to the next step. Or, if you are using jumps, to the next jump selection. The workflow does not terminate.

Mismatched eSOP Task Step and Subprocess Counts

There are more eSOP task steps than template subprocesses in this workflow. The number of subprocesses cannot be less than the number of task steps. If this type of fault occurs, it will cause the workflow to terminate.

Mismatched Linked Document Counts

There are more eSOP linked documents in a step than in its corresponding subprocess. The number of linked documents available for a subprocess cannot be less than the number configured for a task step. If this type of fault occurs, it will cause the workflow to terminate.

Global Subprocess Sample

One global subprocess sample is part of eSOP. You can find this resource in the navigator at **Workflow > Subprocesses**.

Sample Step Template

Demonstrates functionality created for task steps, such as comment forms, emails, keyword replacement, and jumps.

Comment Form Samples

Two jump form samples are part of eSOP. They are build in using the **Designer** tab in Workflow. You can find these resources in the navigator at **Global Displays > Forms**. Forms for making comments allow a run time user to enter text into the displayed form.

Sample Comments Form (Silverlight)

A form based in Silverlight.

Sample Comments Form (WPF)

A form based in WPF.

Jump Samples

Two jump form samples are part of eSOP. They are built in Microsoft[®] Visual Studio. You can find these resources in the navigator at **Global Displays > Forms**. Jump forms indicate that, during run time, the user can jump to the next task step.

Sample Jumps Form (Silverlight)

A form based in Silverlight.

Sample Jumps Form (WPF)

A form based in WPF.

Related reference

[Design eSOP Workflows Overview \(on page 1647\)](#)

[Template Editor \(on page 1641\)](#)

Related information

[Task Construction for Procedures \(on page 1664\)](#)

[Task Maintenance \(on page 1676\)](#)

[Associate a workflow with a template \(on page 1651\)](#)

[Associate a global subprocess with a task step template \(on page 1653\)](#)

Client Layout Configuration for eSOP Users

Client Layout Configuration for eSOP Users

Workflow Designers and Task Builders interact with Workflow differently; therefore, the instructions to set up and access eSOP are different for each user type.

User Access Requirements

Workflow Designers require access to the navigator and **Displays** panel. The eSOP access instructions in this documentation that are specific to these users assume that the client is configured in this manner.

Task Builders require access to only the eSOP **Task Editor** display. The eSOP access instructions in this documentation that are specific to these users assume that the client is configured in this manner, which requires the following conditions to be in place.

1. A *People* resource for the user has been added to the *Personnel* model. For more information, see *Add users and personnel classes*.
2. The user's workstation has been added as a *Computer* resource in the *Proficiency System* model. For more information, see *Add a resource to a model*.
3. A *Group* that is assigned only the default permissions associated with the System Visibility Key Set has been configured in the *Personnel* model. For more information, see *Add Groups* and *Add workstations to key sets for a group*.
4. The user has been designated as a member of the *Group* that is configured for default System Visibility permissions. For more information, see *Add one or more users to a group*.
5. The user's workstation has been configured to show the eSOP solution and **Task Editor** display upon system startup. For more information, see *Configure computer startup options*.
6. The user's workstation has been configured to hide the navigator and the **Display** panel either for:
 - the workstation itself, if the user does not share the workstation with other users who have differing access needs (for more information, see *Configure computer startup options*), or
 - only the user, if the user does share the workstation with other users who have differing access needs (for more information, see *Override a computer's startup options*).

Accessing the Task Editor from the navigator

If you do not want to limit the access of Task Builders to the eSOP Task Editor display only, you must inform them of the following steps to access this display from the navigator.

1. In the navigator, click **eSOP > Tasks**.
2. In the **Displays** panel, click **Task Editor**.

General Controls

When working in eSOP, it is important to save the work you do. If you do not want to save it, you can cancel it.

General Navigation Controls

When you view task information in the **Task Editor**, several controls are available to work with the tasks.

Attribute	Value	Description
Save	N/A	Click to save all changes made in the selected section.
Cancel	N/A	Click to remove all changes made in the selection section.

Task Editor Grid

The **Task Editor** is a starting point for building tasks that will be used in running workflows. Using this editor, you create and manage the configuration data in the task you are building.

Task Information Grid Headings

You can see task information displayed in a grid for easy viewing. To add or modify task information, including adding a task step to a task, click the **Edit** button.

Attributes	Value	Description
Task	User-defined	Specifies a unique name assigned to a task.
Display Name	User-defined	Specifies a name given to a workflow task for the user to see during run time.

Attributes	Value	Description
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.
Group	User-defined	Specifies a group name for managing similar tasks. You can sort tasks based on their group. Click the drop-down arrow to select an existing group.

General Controls

When you view task information in the **Task Editor**, several controls are available to work with the tasks.

Button	Value	Description
Edit OR Tasks	N/A	Click the Edit button to enter the configuration section for a task or task step. Click the Tasks button to view tasks in a table format.
Add	N/A	Click to add a new task.
Duplicate	N/A	Click to re-create an existing task.
Delete	N/A	Click to delete the selected existing task.

Basic Settings for Building Tasks

The **Task Editor** is a starting point for building tasks that will be used in running workflows. Using this editor, you create and manage the configuration data in the task you are building.

Insert More Task Steps

You can add more task steps to a task. The number of steps you can add is predetermined by the template.

Attributes	Value	Description
Add Step	N/A	Click to add a task step to a step.
Delete Step	N/A	Click to remove a task step from the selected step.

Details Tab - General Information

To add data to a task, including adding subprocess task steps, you select a specific task that has already been entered into the system. To view task information in a table, click the **Tasks** button.

Attributes	Value	Description
Name	User-defined	Specifies a unique name assigned to a task.
Display Name	User-defined Workflow level	Specifies a name given to a workflow task for the user to see during run time.
Description	User-defined up to 255 characters	Provides more information about a resource that a user may find helpful.
Expiration	User-defined	Use the up and down arrows to enter an expiration day or time (hours, minutes, seconds).
Priority	User-defined	Specifies the importance of a task by indicating the preferred order that the task should be completed.
Group	User-defined	Specifies a group name for managing similar tasks. You can sort tasks based on their group. Click the drop-down arrow to select an existing group.

Categories

You can filter the Task List by preconfigured categories. These categories organize concepts or features into groups. When you apply category filters, you focus information within your Task List.

Attributes	Value	Description
Set Categories	N/A	Filter tasks or task steps by a predefined grouping.

Template

Click **Set** to associate a workflow template to a task.

Attributes	Value	Description
Template	User-defined	Click the ellipsis button to select a template from the Task Template folder. To remove the selection, click the Clear button.

Assignments

Task assignments include routing the task to a location or personnel resource depending on the requirement of the workflow.

Attributes	Value	Description
Personnel	User-defined	Click the ellipsis button to select the default person that tasks are routed to. To remove the selection, click the Clear button.
Location	User-defined	Click the ellipsis button to select the default location that tasks are routed to. To remove the selection, click the Clear button.

Basic Settings for Building Task Steps

The **Task Editor** is a starting point for building tasks that will be used in running workflows. Using this editor, you create and manage the configuration data in the task you are building.

Details Tab - General Information

To add data to a task step, you select a specific task that has already been entered into the system. To view task information in a table, click the **Tasks** button.

Attributes	Value	Description
Name	User-defined	Specifies a unique name assigned to a task.
Display Name	User-defined Workflow level	Specifies a name given to a workflow task for the user to see during run time.
Expiration	User-defined	Use the up and down arrows to enter an expiration day or time (hours, minutes, seconds).

Categories

You can filter the Task List by preconfigured categories. These categories organize concepts or features into groups. When you apply category filters, you focus information within your Task List.

Attributes	Value	Description
Set Categories	N/A	Filter tasks or task steps by a predefined grouping.

Assignments

By default, subprocess assignments are configured at the workflow level, where you can bind personnel and location properties, as well as write to the assignments from anywhere in the workflow. However, you can override these assignments for a specific subprocess and reassign the tasks to other personnel or locations.

Attributes	Value	Description
Override Assignments	User-defined	Select this check box to allow an operator to override the equipment and/or personnel resources assigned to the task step.

Attributes	Value	Description
Personnel	User-defined	Click the ellipsis button to select the default person that tasks are routed to. To remove the selection, click the Clear button.
Location	User-defined	Click the ellipsis button to select the default location that tasks are routed to. To remove the selection, click the Clear button.

Status

You can choose whether or not a task step executes as part of the task at run time.

Attributes	Value	Description
Enable	User-defined	Click to enable the task step so that it executes as part of the task at run time.
Disable	User-defined	Click to disable the task step so that it does not execute as part of the task at run time.

Task Schedules in eSOP

After you create a task, you can schedule it by designating the event that triggers it. If a task should be performed in response to more than one event, you can create multiple schedules for the task. You can also create a schedule without an associated triggering event, such as when you are configuring a task step for manual launch.

Enabling Schedules

For convenience, you can enable or disable at once all schedules associated with a task. To enable or disable individual schedules, select a schedule, and then go to the **Status** section.

Attributes	Value	Description
Enable All	N/A	Click to activate all schedules associated with the task.

Attributes	Value	Description
Disable All	N/A	Click to deactivate all schedules associated with the task.

Insert More Schedules

You can add more schedules to a task. The number of schedules you can add is predetermined by the template.

Attributes	Value	Description
Add Schedule	N/A	Click to add a schedule to a task.
Delete Schedule	N/A	Click to remove a schedule from a task.

Schedules Tab - General Information

Attributes	Value	Description
Name	User-defined	Specifies a unique name assigned to a task.
Is Schedule Visible in Start Task Window	N/A	Select this check box to ensure that the selected schedule appear in the list of tasks presented when you click the Start Task button.

Trigger Event

You can set an event that will activate the schedule for execution.

Property	Value	Description
Set	User-defined	Click to select a trigger event.
Clear	N/A	Click to remove the selected event.

Status

You can choose whether or not a schedule executes as part of the task at run time.

Attributes	Value	Description
Enable	User-defined	Click to enable the schedule so that it executes as part of the task at run time.
Disable	User-defined	Click to disable the schedule so that it does not execute as part of the task at run time.

Task Emails in eSOP

Email messages can be sent upon initiation and/or expiration of an overall task or individual task steps, as required.

Insert More Emails

You can add more emails to a task.

Attributes	Value	Description
Add Email	N/A	Click to add an email to a task.
Delete Email	N/A	Click to remove an email from a task.

Email Message Requirements

Enter content into each field to configure the email that you want to send.

Property	Value	Description
Subject	User-defined	Specifies the subject heading of the email message.
Event Trigger	N/A	Select the event that activates the email for sending. You can send emails at both the task and task step level. Task Started

Property	Value	Description
		<p>Sends an email when a task starts.</p> <p>Task Expired</p> <p>Sends an email when a task expires.</p> <p>Task Step Started</p> <p>Sends an email when a task step starts.</p> <p>Task Step Expired</p> <p>Sends an email when a task step expires.</p>
Message	User-defined	Specifies the content in the body of the email message.
Recipients	User-defined	To include people who you want to send emails to, click Add Recipient .

Keywords in eSOP

If the Workflow Designer has configured keyword properties for a task or task step, you view them on the **Keywords** tab.

System-defined keywords are marked **Reserved**. User-defined keywords can be added to the list by clicking the **Add** button. All keywords can be removed from the list.

To sort the table alphabetically, click on the column heading.

Property	Value	Description
Name	User-defined OR System-defined	Specifies the names of workflow properties configured for the selected task.

Property	Value	Description
Description	User-defined ORSystem-defined	Specifies more information about the related workflow properties.

Related reference

[Keyword Properties \(on page 1496\)](#)

Related information

[Replace keyword text during run time \(on page 1494\)](#)

Template Editor

You use the **Template Editor** to associate a workflow with a template, customize preconfigured user controls, and indicate specific text to be replaced with live values during workflow run time.

Set Templates

Workflow Designers create workflows to be used as eSOP task templates. You must first define the workflow and configure the template options.

Property	Value	Description
Set	N/A	Click to select an existing workflow.
Template Type	N/A	<p>From the drop-down menu, select one of the following template type to associate with your workflow template.</p> <p>Sequential</p> <p>Sequential templates have sub-processes that are aligned in a linear order and displayed top to bottom; that is, the task steps are all the same. As a result, no branch-</p>

Property	Value	Description
		<p>ing or looping sub-processes are available; however, you can jump between steps by configuring jump options, as well as move task steps around.</p> <p>Fixed</p> <p>With fixed templates, the number of steps in the task cannot be increased from the pre-set amount; however, you can remove steps, add missing steps, and re-order the steps.</p> <p>Parallel</p> <p>In these template types, the workflow contains a Parallel activity, allowing for branched sub-processes.</p>

Custom Configuration Panels

You can configure panels for both the task and task step level. After you add the assembly in the *Config Panel Editor*, you can further manipulate the order of the panels in the **Template Editor**. The outcome of these configurations are displayed in the *Task Editor* and viewed by the person constructing tasks.

Property	Value	Description
Add Property	N/A	Click to add a configuration panel resource from Config Panels . This resource is then displayed as tabs in the Task Editor for the Task Builder.
Remove	N/A	Click to remove a configuration panel resource from the table.
 OR 	N/A	Click the up or down arrow to change the order that the tabs are displayed in the Task Editor for the Task Builder.
Order	User-defined	Click the selected table row to move the tab that appears in the Task Editor moves further to the left in the display.
Tab Name	User-defined	Specifies the name attributed to the configuration panel resource created in Config Panels . To change the name of the tab as it appears in the Task Editor , double-click the table cell, and enter new text.
Configuration	User-defined	Specifies additional custom configuration information for a panel. To add or modify content, double-click the table cell, and then add .csv or .dita text to customize a particular configuration panel instance.
Config Panel	User-defined	Specifies the name of the configuration panel resource from Config Panels .

Custom Template Configuration

Workflow Designers can specify additional custom configuration information for the selected template within the **Template Editor**.

To sort the table alphabetically, click on the column heading.

Property	Value	Description
Name	User-defined	Specifies a name for the type of metadata configured for the selected template.
Metadata	User-defined	Specifies the custom configuration information for the selected template. Maximum metadata value length is 4,000 characters.

Work Instructions

The **Work Instructions** tab is where you can input detailed instructions for a person performing a task or task step.

These instructions appear as a **Work Instructions** button in the Task List. If no information is available, this section is not visible to an operator. The information can include anything that an operator requires to perform a workflow task or a subprocess task step, or finds useful to complete the task/task step.



Note:

If content in the work instructions area exceeds a certain threshold, the editor may not perform optimally during design time, as well as in the Task List during run time. To prevent this issue, use the following recommendations: use small graphic files; use small amounts of text, or several line breaks, within the body of the work instructions area.

Updating Work Instructions During Run Time

You can use keywords to replace text in work instructions and email subjects and bodies. This text is replaced with keyword values in a workflow at run time.

For more information, see *Replace keyword text during run time*.

Attributes

The following table describes the available options.

Attribute	Value	Description
Task/Task Steps list	N/A	Specifies the name of the currently selected workflow, and the names of all subprocesses associated with that workflow.
Work Instructions	User-defined	Select the workflow or a subprocess, and then enter instructions related to that task or task step.

Linked Documents in eSOP

You can link documents or web sites in task steps.

Insert More Documents

You can add more documents to a task step.

Attributes	Value	Description
Add Linked Doc	N/A	Click to add a document to a task step.
Delete Linked Doc	N/A	Click to remove a document from a task step.

Requirements for Linking Documents

Enter content into each field to configure documents that you want to link to a task step.

Property	Value	Description
Display Name	User-defined	Specifies a unique identifier for the document you are linking.
URL/UNC	User-defined	Displays the path to the document that you are linking. You can enter the full path, or select a

Property	Value	Description
		<p>document from a directory. You must always enter an absolute path.</p> <p>The following documents types are supported in Safari: .doc,- .docx, .xls, .xlsx, .ppt, .pptx, .pdf,- .jpg, .png, .gif</p> <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: For URLs, you must enter <code>http://.</code> </div>
Browse	N/A	Click to open the Browse window to locate a document.
Preview	N/A	Click to view the document or web site you inserted.

Jumping Task Steps

You can configure jumps for your task if, during run time, task steps need to be skipped or repeated.

Insert a Jump Action

You can configure a task step to jump to another task step.

Attributes	Value	Description
Add Jump	N/A	Click to add a jump action to a task step.
Delete Jump	N/A	Click to remove a jump action from a task step.

SectionTitle

Property	Value	Description
Name	User-defined	Specifies the name of the jump action.
Jump To	N/A	From the drop-down list, select the name of the task step that you want the selected task step to jump to.
Clear	User-defined	Click to remove the selected task step jump.

Design eSOP Workflows Overview

Design eSOP Workflows Overview

eSOP requires a person with expert knowledge in workflow development to design a workflow before a person with production-specific knowledge can construct task content for the workflow.

A workflow being used in eSOP must be designed for that specific use. A sample workflow template named *Sample Sequential Template* is part of installation. We recommend that you use this type of template as the basis of your eSOP workflow. *To use any sample resource, you must duplicate and re-name it before you modify it.*



Note:

Changing the template associated with a task may have implications for the configuration of the task, such as rendering your task invalid if there are more steps in the template than in the task.

The following information outlines the design options for your workflow.

- Develop a workflow, or use a duplicate version of the sample template.
 - To design a complete workflow, see *Workflow Orientation*.
- In the **Workflow Editor**, customize the duplicated sample template. The following list contains key features that are specific to eSOP workflows.
 - The *Load* and *Preprocess* steps contains the service interface, **IEsopRuntimeService**. This service is used to access eSOP functionality from a workflow, such as:
 - GetExtendedTaskInfoByName** to retrieve the configuration of an eSOP task;
 - JumpToTaskStep** to start a jump in a sequential template.

- The *Presubprocess* step, which runs at the beginning of every subprocess, performs any initialization work required for each subprocess in order to ensure that certain subprocess properties and resources are available during run time.
- eSOP-specific activities. If your workflow contains logic related to linked documents and keyword activities, use the *Read Task Definition* and *Read Task Step Definition* to configure these resources to be available during workflow run time in the *Load* and *Preprocess* steps.

**Note:**

If you are using a copy of the *Sample Sequential Template* from a version older than Workflow 2.5, then you must update your existing workflow in order to access all the new functionality. As such, the new *Categories* property on the *Read Task Definition* and the *Read Task Step Definition* activities must be written to the *Categories* property on the workflow and subprocess respectively.

- To provide more information at run time for the user, you can add *Work Instructions* or *Linked Documents* to a task or task step. Work instructions can be changed using keywords through the *Update Work Instruction Keywords* activity.
- To use email to interact with users when a workflow is running, you can use the **EmailService** service interface to send emails automatically to the appropriate user to indicate when tasks have started or expired. For example, depending on the context, you can use keywords to replace the text in the email subject and/or body to reflect the task being performed.
- In the eSOP model, if you want to add custom forms for use by the Task Builder, add the related assembly in **Config Panels**.
- Using **Task Templates**, associate a workflow with one of three eSOP template for use by Task Builders. You can also add custom forms for task that are available for use by the Task Builder.
- Using **Task Step Templates**, you can associate a predefined global subprocess with your eSOP task template to add to your workflow. You can also add custom forms for task steps that are available for use by the Task Builder.
- Keywords. You can add a list of the available custom keywords that the Task Builder can use when working with tasks. You can add keywords in two ways:
 - Use the **Keywords** tab in the **Template Editor**
 - Use a *Replace Keywords* or *Update Work Instruction Keywords* activity

Related reference

[Sample Files Available with eSOP Installation \(on page 1628\)](#)

[Template Editor \(on page 1641\)](#)

Add custom input panels to create dynamic tabs in the Task Editor

Workflow designers can import a user control that has been designed using third-party software. This control enables additional customized configuration during the task construction phase that task builders use when configuring the final content.

Before you begin



Note:

Controls must have been created in Microsoft® Visual Studio and support the **IConfigPanel** interface. This interface is defined in the `Proficy.eSOP.Interfaces` assembly.

Procedure

1. In the navigator, click **eSOP**.
2. In the **Folders** list, click **Config Panels**, and then click **Add Config Panel**.
3. In the **Name** and **Description** fields, enter a name and description for the resource.
4. Click **OK**.
5. Select the custom panel you just created, or select an existing panel.
6. In the **Displays** panel, click **Config Panel Editor**.
7. Click **Set** next to the **.NET or Silverlight Assembly** field.
The **Open** dialog box appears.
8. Locate and select the assembly file you want to use, and then click **Open**.
The assembly file name appears and the **Class Name** drop-down list is populated with the classes specific to the selected assembly.
9. From the **Class Name** drop-down list, select a class.
10. **Optional:** To change assemblies, click **Cancel**.
11. Click **Save**.
12. In the **Folders** list, click **Task Templates**, and then select a template.
13. In the **Displays** panel, click **Template Editor**.
14. In the **Template Workflow** area, add a task template.
15. You can add custom tabs to the **Task Editor** for both tasks and task steps.
 - To add tabs to tasks, go to the **Task Custom Configuration Panels** area.
 - To add tabs to task steps, go to the **Task Step Custom Configuration Panels** area.

**Tip:**

For viewing preferences, you can move the table columns around. Click and hold the column title and drag it to a new location within the table.

Click this button...	To...
Add	select an existing panel from Config Panels .
Remove	delete a selected panel from the table.
	move the selected custom panel to a higher level in the Task Editor , so that the tab of the custom panel appears closer to the standard tabs. That is, the tab moves further left.
	move the selected custom panel to a lower level in the Task Editor , so that the tab of the custom panel appears further from the standard tabs. That is, the tab moves further right.

16. After you select a custom configuration panel, you can make further modifications related to the panel inside the table.
 - a. To customize the name of the tab when viewed in the **Task Editor**, in the **Tab Name** column, double-click inside the table cell to modify the name.
The new name reflects the tab name that will be viewed by the Task Builder during task construction in the **Task Editor**.
 - b. To include additional custom configuration information for a panel, in the **Configuration** column, double-click inside the table cell to modify the content. You can add .csv or .dita text to customize a particular configuration panel instance.

17. Click **Save**.

Related information

[Associate a workflow with a template \(on page 1651\)](#)

Associate a workflow with a template

Workflow Designers create workflows to be used as eSOP task templates. You must first define the workflow and configure the template options. Then, in **Config Panels**, you use templates for additional, custom configuration.

Before you begin

A sample workflow template named **Sample Sequential Template** is part of installation. We recommend that you use this type of template as the basis of your eSOP workflow.



Note:

All workflows to be used as task templates require the input parameter **TaskName**.

Procedure

1. In the navigator, click **eSOP**.
2. In the **Task Templates** list, click **Task Templates**, and then click **Add Task Template**.
3. In the **Name** and **Description** fields, enter a name and description for the resource.
4. Click **OK**.
5. In the **Displays** panel, click **Template Editor**.
6. In the **Template Workflow** area, click **Set**.
The **Universal Browser** appears.
7. Select an existing workflow, and then click **OK**.
8. In the **Template Type** field, from the drop-down menu, select the type of template that your workflow was created against.



Tip:

You cannot save an invalid task. If you get a validation error message, adjust the number of steps in your task, and then click **Save**.

Select...	If you...
Sequential	created a sequential template, or used the sample template, Sample Sequential Template .
Fixed	want the number and position of task steps in a workflow to be pre-set. With this type, you are unable to configure jump options or move task steps around.

Select...	If you...
Parallel	require a workflow with parallel subprocesses.

9. If you added custom configuration panels that you want to appear for all task and/or task step instances based on this template, perform one of the following actions.

 **Tip:**
For viewing preferences, you can move the table columns around. Click and hold the column title and drag it to a new location within the table.

Click this button...	To...
Add	select an existing panel from Config Panels .
Remove	delete a selected panel from the table.
	move the selected custom panel to a higher level in the Task Editor , so that the tab of the custom panel appears closer to the standard tabs. That is, the tab moves further left.
	move the selected custom panel to a lower level in the Task Editor , so that the tab of the custom panel appears further from the standard tabs. That is, the tab moves further right.

10. After you select a custom configuration panel, you can make further modifications related to the panel inside the table.
- a. To customize the name of the tab when viewed in the **Task Editor**, in the **Tab Name** column, double-click inside the table cell to modify the name.
The new name reflects the tab name that will be viewed by the Task Builder during task construction in the **Task Editor**.
 - b. To include additional custom configuration information for a panel, in the **Configuration** column, double-click inside the table cell to modify the content. You can add .csv or .dita text to customize a particular configuration panel instance.

11. To add custom keywords, on the **Keyword** tab, click **Add**.

- a. In the **Name** column, double-click inside the table cell to modify the name.
 - b. In the **Description** column, double-click inside the table cell to enter information that describes the new keyword.
 - c. **Optional:** To delete an existing keyword, select it, and then click **Remove**.
12. To add custom configuration information, on the **Metadata** tab, click **Add**.
- a. In the **Name** column, double-click inside the table cell to modify the name.
 - b. In the **Metadata** column, double-click inside the table cell to enter a value that has a maximum length of 4000 characters.
 - c. To delete an existing row, select it, and then click **Remove**.
13. Click **Save**.

Associate a global subprocess with a task step template

Workflow designers create global subprocesses to be used as eSOP task step templates. You must first define the workflow subprocesses and configure the template options. Then, in **Config Panels**, you use task templates for additional, custom configuration.

Procedure

1. In the navigator, click **eSOP**.
2. In the **Task Step Templates** list, click **Task Step Templates**, and then click **Add Task Step Template**.
3. In the **Name** and **Description** fields, enter a name and description for the resource.
4. Click **OK**.
5. Select the template you just created, or select an existing one.
6. In the **Displays** panel, click **Template Editor**.
7. In the **Template Subprocess** area, click **Set**.
The **Universal Browser** appears.
8. Select an existing subprocess, and then click **OK**.
9. If you configured custom configuration panels that you want to add to this subprocess task step, perform one of the following actions.

Click this button...	To...
Add	select an existing panel from Config Panels .
Remove	delete a selected panel from the table.
	move the selected custom panel to a higher level in the Task Editor , so that the tab of the cus-

Click this button...	To...
	tom panel appears closer to the standard tabs. That is, the tab moves further left.
	move the selected custom panel to a lower level in the Task Editor , so that the tab of the custom panel appears further from the standard tabs. That is, the tab moves further right.

10. Click **Save**.

Define available keyword text substitutions during run time

Keywords are derived from local variables and properties configured in a workflow. Keywords enable the creation of parameterized content, such as work instructions and email, with specific text replaced with live values during workflow run time. All specially delimited keywords in a document or text string are replaced with keyword values in a workflow at runtime.

Prerequisites

[Associate a workflow with a template \(on page 1651\)](#)

Procedure

1. In the navigator, click **eSOP > Task Templates**.
2. In the resource list, add a resource, or select the resource that you want to work with.
3. In the **Displays** panel, click **Template Editor**.
4. To add custom keywords, on the **Keyword** tab, click **Add**.
 - a. In the **Name** column, double-click inside the table cell to modify the name.
 - b. In the **Description** column, double-click inside the table cell to enter information that describes the new keyword.
 - c. To delete an existing keyword, select it, and then click **Remove**.
5. Click **Save**.

Related reference

[Keyword Properties \(on page 1496\)](#)

[Config Panel Editor \(on page 1656\)](#)

Replace keyword text during run time

You can use keywords to replace text in work instructions and email subjects and bodies. This text is replaced with keyword values in a workflow at run time.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Drag and drop a **Local Subprocess** into the workflow and double-click it, or double-click an existing subprocess.



Note:

If required, configure the workflow.

6. There are two types of keyword activities you can use.

Insert...	To...
Replace Keywords	replace keywords in a block of text. Go to step 7.
Update Work Instruction Keywords	ensure that the change in the value of a workflow variable is reflected in the work instruction text. Go to step 8.

7. To use the Replace Keywords activity, perform the following procedure.
 - a. Insert a Replace Keywords activity.
 - b. In the **Keyword Text** field, bind the activity to the property whose text you want to replacement .
The activity's output is the text that was defined to replace the keyword.
8. To automatically update keywords in work instructions when the workflow begins, insert the Update Work Instruction Keywords activity.



Note:

This activity is used only if the values in a work instruction should be updated during the execution of a task step.

9. Click **Save**.

Add custom configuration to a template

Workflow Designers can specify additional custom configuration information for the selected template within the **Template Editor**.

Prerequisites

[Associate a workflow with a template \(on page 1651\)](#)

Procedure

1. In the navigator, click **eSOP > Task Templates**.
2. In the resource list, add a resource, or select the resource that you want to work with.
3. In the **Displays** panel, click **Template Editor**.
4. To add custom configuration information, on the **Metadata** tab, click **Add**.
 - a. In the **Name** column, double-click inside the table cell to modify the name.
 - b. In the **Metadata** column, double-click inside the table cell to enter a value that has a maximum length of 4000 characters.
 - c. To delete an existing row, select it, and then click **Remove**.
5. Click **Save**.

Related reference

[Template Editor \(on page 1641\)](#)

[Custom Template Configuration \(on page 1644\)](#)

Config Panel Editor

You use the **Config Panel Editor** to create custom configuration panels. These panels are based on existing forms or user displays that have been created in Workflow.

Property	Value	Description
.NET or Silverlight Assembly	User-defined	Click Set to locate and select the assembly that contains the display you want to add.
Class Name	User-defined	Specifies the class name of the display contained by the selected assembly.

Template Editor

You can configure panels for both the task and task step level. After you add the assembly in the **Config Panel Editor**, you can further manipulate the order of the panels in the **Template Editor**. The outcome of these configurations are displayed in the **Task Editor** and viewed by the person constructing tasks.

Related reference

[Template Editor \(on page 1641\)](#)

Related information

[Add custom input panels to create dynamic tabs in the Task Editor \(on page 1649\)](#)

Replace Keywords Activity

You can use keywords to replace text in email subjects and bodies. This text is replaced with keyword values in a workflow at run time.

Inputs

Inputs	Value	Description
Keyword Text	String	Specifies the text that you want to replace. Use the following format: <@Keyword@>.

Outputs

Outputs	Value	Description
Result Text	String	Displays the text that has been replaced with the keyword value.

Related reference[Keyword Properties \(on page 1496\)](#)**Related information**[Replace keyword text during run time \(on page 1494\)](#)

Update Work Instruction Keywords Activity

This activity updates task step work instructions with current keyword values during run time.

You use this activity if the value of a work instruction requires updating during the execution of a task step. Then, the work instruction is automatically updated when the workflow begins and any task step begins.

Keyword Properties

Keywords are derived from local variables and properties configured in a workflow. Keyword property values are substituted during run time.

You can add keywords in two ways:

- Use the **Keywords** tab in the **Template Editor**
- Use a *Replace Keywords* or *Update Work Instruction Keywords* activity

You can add keywords using a *Replace Keywords* or *Update Work Instruction Keywords* activity in a workflow.

Additionally, custom keywords can be defined as *local variables* within a workflow or subprocess with the variable name referenced directly inside the standard keyword delimiters. Keywords are resolved at the lowest possible scope. For example, if a workflow and a subprocess both have a **Name** property, the `<@Name@>` keyword in a subprocess will be substituted with the subprocess name since that is the immediate scope, not the workflow name (the parent scope).

**Note:**

The list of keywords associated with a task template is a reference available for Task Builders. It does not make a keyword value available during run time unless the keyword name maps to an existing property or local variable.

Workflow Properties

For a list of properties that can be used with keywords, see *Workflow properties*.

Subprocess Properties

For a list of properties that can be used with keywords, see *Properties for subprocesses*.

Read Task Definition Activity

You can use this activity anywhere within a workflow if you need to read a configuration that is specific for use with eSOP.

Inputs

Processes an eSOP task and returns all of the task's properties.

Inputs	Value	Description
Task Info Extended	Object	Specifies the task information returned by a GetExtendedTaskInfo method call.

Outputs

The following properties are associated with the task being processed.

Inputs	Value	Description
ID	GUID	Specifies a unique identifier for the task being processed.
Name	String	Specifies a unique name assigned to a task.
Description	String	Specifies the description provided for the task being processed.
Group	String	Specifies a group name for managing similar tasks. You can sort tasks based on their group. Click the drop-down arrow to select an existing group.
Display Name	String	Specifies a name given to a workflow task for the user to see during run time.

Inputs	Value	Description
Location Assignment	DirectoryResource	Specifies the assigned location of the task being processed.
Personnel Assignment	DirectoryResource	Specifies the assigned human resource of the task being processed.
Enabled	Boolean	Specifies the activation status of the task being processed. True if enabled, False if disabled.
Priority	Int32	Specifies the importance of the task being processed, as determined during configuration.
Expiration	TimeSpan	Specifies the time that the task being processed is set to expire, as determined during configuration.
Steps	List<TaskStepInfoExtended>	Specifies the task steps associated with the task being processed.
Started Emails	List<Email>	Specifies the emails that will be sent, as determined during configuration, when the task being processed starts.
Expiry Emails	List<Email>	Specifies the emails that will be sent, as determined during configuration, when the task being processed expires.
Config Panel Data	List<ConfigPanelDatumInfo>	Specifies all data provided by custom configuration panels for the task.
Categories	CategoryList	Specifies all the categories related to the task being processed.

Read Task Step Definition Activity

You can use this activity anywhere within a workflow if you need to read configuration that is specific for use with eSOP.

Inputs

Processes an eSOP task step and returns all of the task step's properties.

Inputs	Value	Description
Task	Object	The task information returned by a GetExtendedTaskInfo method call.
Subprocess Id	String	Specifies the unique identifier of the subprocess for which you want step configuration.

Outputs

The following properties are associated with the task step being processed.

Inputs	Value	Description
ID	GUID	Specifies a unique identifier for the task step being processed.
Name	String	Specifies a name assigned to the task step being processed.
Display Name	String	Specifies a name given to a workflow task step for the user to see during run time.
Location Assignment	DirectoryResource	Specifies the assigned location of the task step being processed.
Personnel Assignment	DirectoryResource	Specifies the assigned human resource of the task step being processed.

Inputs	Value	Description
Sequence	Int32	Specifies the order within the workflow that the task step is processed.
Enabled	Boolean	Specifies the activation status of the task step being processed. True if enabled, False if disabled.
Expiration	TimeSpan	Specifies the time that the task step being processed is set to expire, as determined during configuration.
Linked Documents	List<LinkedDocInfo>	Specifies the documents that have been linked to the task step being processed during configuration.
Started Emails	List<Email>	Specifies the emails that will be sent, as determined during configuration, when the task step being processed starts.
Expiry Emails	List<Email>	Specifies the emails that will be sent, as determined during configuration, when the task step being processed expires.
Jumps	List<JumpInfo>	Specifies the jumps associated with the task step being processed, as determined during configuration.
Config Panel Data	List<ConfigPanelDatumInfo>	Specifies all data provided by custom configuration panels for the task step.
Categories	CategoryList	Specifies all the categories related to the task step being processed.

Sequential Templates

Sequential templates have subprocesses that are aligned in a linear order and displayed top to bottom; that is, the task steps are all the same. As a result, no branching or looping subprocesses are available; however, you can jump between steps by configuring jump options, as well as move task steps around.

**Note:**

The number of steps in the task must be less than or equal to the number of steps in the template; otherwise, the task is invalid.

Related reference

[Sample Files Available with eSOP Installation \(on page 1628\)](#)

Related information

[Create and link a task \(on page 1665\)](#)

[Change template association \(on page 1677\)](#)

Parallel Templates

In these template types, the workflow contains a Parallel activity, allowing for branched subprocesses.

**Note:**

The number of steps in the task must be less than or equal to the number of steps in the template; otherwise, the task is invalid.

Fixed Templates

With fixed templates, the number of steps in the task cannot be increased from the pre-set amount; however, you can remove steps, add missing steps, and re-order the steps.

Usage Restrictions

The following restrictions apply when using fixed templates:

- If the number of steps in the task is more than the number of steps in the template, the task is invalid.
- If the template is changed or edited, and the number of steps is less than the number of subprocesses, new steps are not added automatically. The user must add new steps to make the task compatible with the modified template. However, if you change a template to another template that includes no task steps, steps are added automatically.



Note:

If the steps in the fixed template do not match the corresponding subprocesses in the template workflow (that is, they have been moved, added, or deleted), the task steps in the **Task Editor** are highlighted in yellow. If the linked steps no longer exist in the corresponding template workflow, the task steps are highlighted in red.

Task Construction for Procedures

Task Construction for Procedures

After templates have been developed for the various types of standard operating procedures in your organization, you can construct tasks to meet your specific operational requirements.

Task construction in eSOP involves configuration at both the task level and at the level of each step within a task.

The order in which you perform the steps to construct tasks may vary somewhat from the order presented in this documentation. For example, whereas you must create a task and associate it with a template and with personnel and/or equipment before creating task steps, you can create task steps before configuring email messages for the task itself.

Likewise, whereas certain steps (such as creating a task) are mandatory, others (such as linking related documents to task steps) may be optional based the specific scenario that the task or task step is designed to address.

Finally, additional construction steps may be required based on the specific needs of your organization as reflected by custom configurations implemented in the related templates.

Related reference

[Design eSOP Workflows Overview \(on page 1647\)](#)

Related information

[Task Maintenance \(on page 1676\)](#)

Create and link a task

The first step in task construction is to create the task and then associate it with the applicable task template, as well as the applicable personnel and/or equipment. As part of this process, you can also specify other details related to the task as a whole, including expiration time and priority.

Procedure

1. In **Task Editor** grid, click **Add**.

**Note:**

If the grid interface is not displayed, click the **Task** button.

2. On the **Details** tab, in the **Template** area, click **Set**.

The **Universal Browser** appears.

3. Select **Task Templates**, and then select the specific template that you want to associate the task with.

**Note:**

For information and restrictions related to task templates, see these topics:

- *Fixed Templates*
- *Parallel Templates*
- *Sequential Templates*

4. Click **OK**.
5. On the **Details** tab, associate human and/or equipment resources with the task.

If you want to...	Do this...
<p style="text-align: center;">associate the task with one or more human resources</p>	<ol style="list-style-type: none"> a. In the Personnel area, click Set. The Universal Browser appears. b. Select Personnel, then select either People (for a single individual) or Classes

If you want to...	Do this...
	(for a group of individuals), and then select the specific person or class. c. Click OK .
associate the task with an equipment resource	a. In the Personnel area, click Set . The Universal Browser appears. b. Select Instances , navigate through the equipment model hierarchy, and then select the specific equipment resource. c. Click OK .



Note:

You must establish the template as well as the personnel and/or equipment associations and save your input before you can configure steps for the task.

6. In the **Name** field, enter the name that the system uses to identify the task.
7. In the **Display Name** field, enter the name that you want displayed in the Task List to identify the task at run time.



Note:

For this and any other field containing text that may be displayed in the Task List, you can type Keywords as part of your entry. For more information, see *Use keywords in eSOP task text*.

8. In the **Description** field, enter text to describe the task.
9. **Optional:** In the **Expiration** field, enter a value or use the arrow to indicate the time span when a task expires.
10. **Optional:** In the **Priority** field, enter the value that indicates the task's relative importance for sorting purposes when multiple tasks are listed in the Task List at run time.



Note:

You can enter whole numbers only.

11. In the **Group** field, select the option that reflects the category that you want to assign the task to, or, if none of the options is appropriate, enter a new value directly in the field.

**Note:**

Although technically optional, this measure facilitates better viewing of tasks in the **Task Editor** grid interface by letting you hide or display task listings by group.

12. Click **Save**.

Related reference

[Fixed Templates \(on page 1663\)](#)

[Parallel Templates \(on page 1663\)](#)

[Sequential Templates \(on page 1663\)](#)

Related information

[Use keywords in eSOP task text \(on page 1687\)](#)

Configure category filtering in eSOP

You can filter the Task List by preconfigured categories. These categories organize concepts or features into groups. When you apply category filters, you focus information within your Task List.

Procedure

1. In the navigator, click **eSOP Tasks Your Task Task Editor**.
2. Click the **Details** tab.
3. Select the task or task step that you want to work with.
4. In the **Categories** section, click **Set Categories** to select a preconfigured category.
The **Universal Browser** appears.
5. Select one or more categories, and then click **OK**.
The selected items are listed in the **Categories** section.
6. **Optional:** To remove a category, click **X**.

Configure task emails

You can configure email messages to be sent upon initiation and/or expiration of an overall task or individual task steps, as required.

Before you begin

To use the email feature in eSOP, you must perform steps 1-4 in the Send emails using the email service provider topic.

Prerequisites

[Create and link a task \(on page 1665\)](#)

[Add task steps \(on page 1669\)](#)

About this task



Note:

If the **Task Editor** workspace displays the grid instead of the task editing interface, select the listing for the task that you want to work on, and then click the **Edit** button. To return to the grid interface, click the **Tasks** button.

Procedure

1. In the **Task Editor**, click the **Email** tab.
2. Select the task listing or the listing for the task step that you want to configure an email for.
3. Click **Add Email**.
4. In the **Subject** field, enter the subject of the email.



Note:

Here, as well as in the email message itself (step 6), you can type *Keywords* as part of your entry. For more information, see [Use keywords in eSOP task text \(on page 1687\)](#).

5. In the **Event Trigger** field, select the option applicable to the configuration scenario.

To send the email when the...	Select this option...
task initiates	Task Started
task expires	Task Expired
task step initiates	Task Step Started
task step expires	Task Step Expired

6. In the **Message** field, type the body of the email.
7. In the **Recipients** area, perform the following procedure.
 - a. Click **Add Recipient**.
The **Universal Browser** appears.
 - b. Select **Personnel**, then select either **People** (for a single individual) or **Classes** (for a group of individuals), and then select the specific person or class.

- c. Click **OK**.
 - d. Repeat steps a-c for each personnel resource designated as a recipient.
8. Click **Save**.
 9. To configure additional emails, perform one of the following procedures.
 - For the currently selected task, repeat steps 3-8.
 - For a task other than the currently selected one, repeat steps 2-8.

Add task steps

After you have created and saved a task, you can create the individual steps for the task. As part of this process, you can also specify other details related to the step, including expiration time, and you can override task-level personnel and equipment associations, as required.

About this task

**Note:**

If the **Task Editor** workspace displays the grid instead of the task editing interface, select the listing for the task that you want to work on, and then click the **Edit** button. To return to the grid interface, click the **Tasks** button.

Procedure

1. In the **Task Editor** workspace, click **Add Step**.

**Note:**

If the **Add Step** button is inactive, either you have not established the required associations for the task and saved your input (see [Create and link a task \(on page 1665\)](#)), or the maximum number of steps that can be configured based on the associated template (as noted under the pane that lists the task steps) has been reached.

2. On the **Details** tab, in the **Name** field, enter the name that the system uses to identify the task step.
3. In the **Display Name** field, enter the name that you want to be displayed in the Task List to identify the task step at run time. Do not use the same display name more than once.



Note:

For this and any other field containing text that may be displayed in the Task List, you can type Keywords as part of your entry.

4. **Optional:** In the **Expiration** field, use the arrow control or directly enter a value to specify the duration after which the task step should be considered expired.
5. **Optional:** If the personnel and/or equipment associated with the task are different for the task step, select the **Override Assignments** check box, and then in the **Assignment** area, change the associations.

If you want to...	Do this...
<p>change personnel resources</p>	<ol style="list-style-type: none"> a. In the Personnel area, click Set. <p>The Universal Browser appears.</p> <ol style="list-style-type: none"> b. Select Personnel, then select either People (for a single individual) or Classes (for a group of individuals), and then select the specific person or class. c. Click OK.
<p>change the equipment resource</p>	<ol style="list-style-type: none"> a. In the Personnel area, click Set. <p>The Universal Browser appears.</p> <ol style="list-style-type: none"> b. Select Instances, navigate through the equipment model hierarchy, and then select the specific equipment resource. c. Click OK.

6. To activate the step as part of the task, in the **Status** area, click the **Enable** button.
7. Click **Save**.
8. For each step you want to add to the task, repeat steps 1 through 7.

Related information

[Create and link a task \(on page 1665\)](#)

[Use keywords in eSOP task text \(on page 1687\)](#)

Configure work instructions for task steps

For each step in a task, you can configure work instructions to be displayed in the Task List at run time.

Prerequisites

[Add task steps \(on page 1669\)](#)

About this task



Note:

If the **Task Editor** workspace displays the grid instead of the task editing interface, select the listing for the task that you want to work on, and then click the **Edit** button. To return to the grid interface, click the **Tasks** button.

Procedure

1. In the **Task Editor**, click the **Work Instructions** tab.
2. Select the listing for the task step that you want to define work instructions for.
3. In the work instructions workspace, enter text.



Note:

For this and any other field containing text that may be displayed in the Task List, you can type Keywords as part of your entry. For more information, see *Use keywords in eSOP task text*. You can also insert information copied from another application, as well as graphics.

4. Click **Save**.
5. For each step that you want to define work instructions for, repeat steps 3-4.

Related information

[Use keywords in eSOP task text \(on page 1687\)](#)

Link external documents to task steps

For each step in a task, you can add external documents to be available from the Task List during run time. Linked documents are only available to users once a task step starts running.

Prerequisites

[Create and link a task \(on page 1665\)](#)

About this task



Note:

If the **Task Editor** workspace displays the grid instead of the task editing interface, select the listing for the task that you want to work on, and then click the **Edit** button. To return to the grid interface, click the **Tasks** button.

Procedure

1. In the **Task Editor**, click the **Linked Documents** tab.
2. Select the listing for the task step that you want to specify related external documents for.
3. Click **Add Linked Doc**.



Note:

You can only add the number of documents that the related workflow (associated template) allows. You can view the maximum number at the bottom of the list pane. If the **Add Linked Doc** button is inactive, the maximum number of documents has been reached.

4. In the **Display Name** field, enter the name that you want to be displayed in the Task List to identify the linked document at run time.



Note:

For this and any other field containing text that may be displayed in the Task List, you can type Keywords as part of your entry. For more information, see *Use keywords in eSOP task text*.

5. Near the **URL/UNC** field, click **Browse**, and then in the displayed dialog box, navigate to and select the document.



Note:

If you know the absolute path or full web site address for the document, you can enter it directly in the field instead.

6. **Optional:** To view the document or web site, click **Preview**.
7. Click **Save**.
8. To add additional external documents, perform one of the following procedures.

- For the currently selected task step, repeat steps 3 through 7.
- For a task step other than the currently selected one, repeat steps 2 through 7.

Related information

[Use keywords in eSOP task text \(on page 1687\)](#)

Configure task step emails

You can configure email messages to be sent upon initiation and/or expiration of task steps, as required.

For the steps to configure task-step emails, see *Configure task emails*.

Related information

[Configure task emails \(on page 1667\)](#)

Define jumps for task steps

You can configure jumps for your task if, during run time, task steps need to be skipped or repeated.

Prerequisites

[Add task steps \(on page 1669\)](#)

About this task



Note:

If the *Task Editor* workspace displays the grid instead of the task editing interface, select the listing for the task that you want to work on, and then click the **Edit** button. To return to the grid interface, click the **Tasks** button.

Procedure

1. In the **Task Editor**, click the **AddJump** tab.
2. Select the listing for the task step that you want to define a jump for.
3. Click the **AddJump** button.
4. In the **Name** field, enter the name that you want displayed in the Task List to identify the jump step at run time.



Note:

For this and any other field containing text that may be displayed in the Task List, you can type Keywords as part of your entry.

5. In the **Jump To** field, select the target task step for the jump.



Note:

If you want to remove the current jump configuration, click **Clear**.

6. Click **Save**.
7. To configure additional jumps, perform one of the following procedures.
 - For the currently selected task step, repeat steps 3-6.
 - For a task step other than the currently selected one, repeat steps 2-6.

Related information

[Use keywords in eSOP task text \(on page 1687\)](#)

Schedule a task

After you create a task, you can schedule it by designating the event that triggers it. If a task should be performed in response to more than one event, you can create multiple schedules for the task.

Prerequisites

[Create and link a task \(on page 1665\)](#)

About this task



Note:

If the **Task Editor** workspace displays the grid instead of the task editing interface, select the listing for the task that you want to work on, and then click the **Edit** button. To return to the grid interface, click the **Tasks** button.

Procedure

1. In the **Task Editor**, click the **Schedules** tab.
2. Click **Add Schedule**.
3. In the **Name** field, enter the name that the system uses to identify the schedule.

4. **Optional:** To manually launch the task at run time using the **Start Task** button in the Task List, select the **Is schedule visible in Start Task window** check box.
5. **Optional:** In the **Trigger Event** area, click **Set** dialog box; select the event trigger you want to use; and then click **Finish**.

**Note:**

You can create a schedule without an associated triggering event, such as when you are configuring a task step for manual launch (see step 4).

6. To activate the schedule, in the **Status** area, click **Enable**.

**Attention:**

Because enabling a schedule activates the task for use at run time, schedules are initially disabled by default, and you should not enable schedules until task configuration is complete. For more information, see *Activate a task*.

7. Click **Save**.
8. To configure additional schedules for the task, repeat steps 2-7.

Related information

[Activate a task \(on page 1675\)](#)

Activate a task

When task construction is complete, you activate the task for use in your production run time environment by enabling one or more schedules associated with the task.

About this task**Note:**

If the **Task Editor** workspace displays the grid instead of the task editing interface, select the listing for the task that you want to work on, and then click the **Edit** button. To return to the grid interface, click the **Tasks** button.

Procedure

1. In the **Task Editor**, click the **Schedules** tab.
2. Activate the task.

If you want to activate the task in relation to...	Do this...
all associated schedules	Click the Enable All button.
individual associated schedules	Select the schedule listing, and then in the Status area, click Enable .

3. Click **Save**.

The task is activated.

Task Maintenance

Task Maintenance

After initially constructing tasks, you can view basic configuration information for all tasks in a single interface, as well as perform a number of maintenance activities.

Task maintenance in eSOP can take place at both the task level and at the level of each step within a task.

View eSOP tasks

After constructing tasks in eSOP, a listing for each task is displayed by group in the grid interface of the **Task Editor**. As part of each task listing, the grid displays basic configuration information, such as display name, description, and group name.

Procedure

1. If the **Task Editor** grid is not displayed, click the **Tasks** toggle button.
2. Navigate the grid to find the task listings to view.

If you want to...	Do this...
view a task listing that is currently hidden	Expand the group that the task belongs to by clicking the downward-pointing arrow beside the group listing.
arrange task listings within groups based on grid column	Click the column heading. The task listings within each group in the grid are sorted in ascending order based on the value in the selected grid column.

If you want to...	Do this...
	<div data-bbox="873 275 1425 436">  Tip: To sort the task listings in descending order, click the column heading again. </div>
<p data-bbox="391 464 732 495">show or hide all task listings</p>	<p data-bbox="857 464 1409 541">Click the double-headed arrow toggle button in the grid heading row.</p> <p data-bbox="857 583 1417 701">Each group in the grid either expands to show all task listings, or collapses to hide all task listings.</p> <div data-bbox="873 737 1425 1136">  Note: When you collapse all task listings, group listings are still displayed. If any of your tasks has not been assigned to a specific group, a blank group listing is displayed in the grid, under which the non-categorized task listings can be found. </div>

Change template association

After initially linking a task with a template, you can change the association.

Procedure

1. In the **Task Editor** grid, click the listing for the task you want to modify.



Tip:

If the grid interface is not displayed, click the **Task** button. If you do not see the task listing in the grid, expand the group that the task belongs to by clicking the downward-pointing arrow beside the group listing.

2. Click the **Edit** button.

The editing interface appears in the **Task Editor** workspace.

3. On the **Details** tab, in the **Template** area, click **Set**.



Note:

Changing the template associated with a task may have implications for the configuration of the task, such as rendering your task invalid if there are more steps in the template than in the task.

4. Click **OK**.

The **Universal Browser** appears.

5. Select **Task Templates**, and then select the specific template that you want to associate the task with.



Note:

For information and restrictions related to task templates, see these topics:

- *Fixed Templates*
- *Parallel Templates*
- *Sequential Templates*

6. Click **OK**.

7. Click **Save**.

Related reference

[Fixed Templates \(on page 1663\)](#)

[Parallel Templates \(on page 1663\)](#)

[Sequential Templates \(on page 1663\)](#)

Change the personnel or equipment associated with a task

After initially linking a task with personnel and/or equipment, you can alter or break association(s).

Procedure

1. In the **Task Editor** grid, click the listing for the task you want to modify.



Tip:

If the grid interface is not displayed, click the **Task** button. If you do not see the task listing in the grid, expand the group that the task belongs to by clicking the downward-pointing arrow beside the group listing.

2. Click the **Edit** button.

The editing interface appears in the **Task Editor** workspace.

3. On the **Details** tab, in the **Assignment** area, change the personnel and/or equipment associated with the task.

If you want to...	Do this...
<p>change the personnel</p>	<p>a. In the Personnel area, click Set.</p> <p>The Universal Browser appears.</p> <p>b. Select Personnel, then select either People (for a single individual) or Classes (for a group of individuals), and then select the specific person or class.</p> <p>c. Click OK.</p>
<p>change the equipment</p>	<p>a. In the Location area, click Set.</p> <p>The Universal Browser appears.</p> <p>b. In the Location area, click Set.</p> <p>c. Click OK.</p>
<p>break the association between the task and personnel or equipment</p>	<p>a. In either the Personnel or Location area, click Clear.</p> <div data-bbox="938 1203 1422 1425" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note: The task must be associated with either a personnel or equipment resource.</p> </div>

4. Click **Save**.

Adjust task scheduling

After initially scheduling a task, you can change the triggering event for a pre-existing schedule, designate additional events that should trigger the task, and/or break the association between an event and the task.

Procedure

1. In the **Task Editor** grid, click the listing for the task you want to modify.



Tip:

If the grid interface is not displayed, click the **Task** button. If you do not see the task listing in the grid, expand the group that the task belongs to by clicking the downward-pointing arrow beside the group listing.

2. Click the **Edit** button.

The editing interface appears in the **Task Editor** workspace.

3. Click the **Schedules** tab.
4. Adjust task scheduling.

If you want to...	Do this...
<p>change the triggering event for a pre-existing schedule</p>	<ol style="list-style-type: none"> a. Select the schedule listing. b. In the Trigger Event area, click Set. <p>The Select Trigger Event dialog box appears.</p> <ol style="list-style-type: none"> c. Select the event trigger you want to use, and then click Finish. <div data-bbox="954 1192 1002 1241" data-label="Image"> </div> <p>Note:</p> <p>You can also change the name that the system uses to identify the schedule (see below) and/or change the setting to control whether the task can be manually launched at run time (see below).</p>

If you want to...	Do this...
	<p>List, select the Is schedule visible in Start Task window check box.</p> <p>d. In the Trigger Event area, click Set.</p> <p>The Select Trigger Event dialog box appears.</p> <p>e. Select the event trigger you want to use, and then click Finish.</p> <p>f. To activate the schedule, in the <i>Status</i> area, click Enable.</p> <div data-bbox="938 730 1422 1178" style="border: 1px solid #f9c77c; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Attention:</p> <p>Because enabling a schedule activates the task for use at run time, schedules are initially disabled by default, and you should not enable schedules until task configuration is complete. For more information, see <i>Activate a task</i>.</p> </div>
<p>break the association between an event and the task</p>	<p>a. Select the schedule configured for the event that you no longer want to trigger the task.</p> <p>b. Click Delete Schedule.</p>

5. Click **Save**.

Related information

[Activate a task \(on page 1675\)](#)

Delete task emails or email recipients

After configuring email messages to be sent upon initiation and/or expiration of an overall task or individual task steps, you can remove personnel from the recipient list or delete the emails altogether.

Procedure

1. In the **Task Editor** grid, click the listing for the task you want to modify.



Tip:

If the grid interface is not displayed, click the **Task** button. If you do not see the task listing in the grid, expand the group that the task belongs to by clicking the downward-pointing arrow beside the group listing.

2. Click the **Edit** button.

The editing interface appears in the **Task Editor** workspace.

3. Click the **Email** tab.
4. Select the listing for the email that you want to delete or remove recipients from.
5. Perform one of the following action(s).

If you want to...	Do this...
<p>delete the email</p>	<p>a. Click the Delete Email button.</p> <p>The email listing disappears from the tab.</p>
<p>remove personnel from the email recipient list</p>	<p>a. In the Recipients area, select the listing for the personnel resource that you want to remove from the recipient list for the email.</p> <p>b. Click the Delete Recipient button.</p> <div data-bbox="941 1318 1422 1675" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note:</p> <p>At least one recipient must be designated per email message. For instructions on adding email recipients and configuring other email settings, see <i>Configure task emails</i>.</p> </div>

6. Click **Save**.

Related information

[Configure task emails \(on page 1667\)](#)

Remove a task step

If an action is no longer applicable to a procedure, you can remove the corresponding step from the task configured for the procedure.

About this task



Note:

If a step is the target of a jump that has been configured to facilitate skipping or repeating steps under certain conditions, the jump must be deleted before the step can be removed.

Procedure

1. In the **Task Editor** grid, click the listing for the task you want to modify.



Tip:

If the grid interface is not displayed, click the **Task** button. If you do not see the task listing in the grid, expand the group that the task belongs to by clicking the downward-pointing arrow beside the group listing.

2. Click the **Edit** button.

The editing interface appears in the **Task Editor** workspace.

3. Select the listing for the step that you want to remove from the task.

4. Click the **Delete Step** button.

A dialog box appears, cautioning you that proceeding will result in permanent deletion of the step and all related configuration data.

5. Click **OK**.

The step listing is permanently removed.

6. Click **Save**.

Disassociate an external document and a task step

If an external document that was previously linked to a task step is no longer applicable to corresponding action, you can break the link between the two items.

Procedure

1. In the **Task Editor** grid, click the listing for the task you want to modify.



Tip:

If the grid interface is not displayed, click the **Task** button. If you do not see the task listing in the grid, expand the group that the task belongs to by clicking the downward-pointing arrow beside the group listing.

2. Click the **Edit** button.

The editing interface appears in the **Task Editor** workspace.

3. Select the listing for the step that you want to disassociate the external document from.
4. Click the **Linked Documents** tab.
5. Select the listing for the external document that you want to disassociate from the step.
6. Click **Delete Linked Doc.**

The document listing is permanently removed.

7. Click **Save**.

Delete task step jumps

If a jump that was previously configured to facilitate skipping or repeating task steps is no longer applicable, you can delete it.

Procedure

1. In the **Task Editor** grid, click the listing for the task you want to modify.



Tip:

If the grid interface is not displayed, click the **Task** button. If you do not see the task listing in the grid, expand the group that the task belongs to by clicking the downward-pointing arrow beside the group listing.

2. Click the **Edit** button.

The editing interface appears in the **Task Editor** workspace.

3. Click the listing for the step that the jump was configured for.
4. Click the listing for the jump that you want to delete.
5. Click the **Delete Jump** button.

The jump listing is permanently deleted.

6. Click **Save**.

Duplicate a task

If you want to use a previously created task as the basis for another task, you can duplicate the original task, and then modify the new task as required.

About this task

**Note:**

When you duplicate a task, the schedules associated with the original task are not duplicated for the task copy.

Procedure

1. In the **Task Editor** grid, click the listing for the task you want to modify.

**Tip:**

If the grid interface is not displayed, click the **Task** button. If you do not see the task listing in the grid, expand the group that the task belongs to by clicking the downward-pointing arrow beside the group listing.

2. Click the **Duplicate** button.

The editing interface appears in the **Task Editor** workspace.

3. Define the applicable schedules for the task, and then update other task configuration settings, as required. For more information, see *Task Construction for Procedures* and *Task Maintenance*.

Related information

[Task Construction for Procedures \(on page 1664\)](#)

[Task Maintenance \(on page 1676\)](#)

Disable a task step

If a task step is not currently applicable to a procedure, you can disable the step so that it does not execute as part of the task at run time. You can see the activation state under the step name in the task list pane.

About this task

**Note:**

If a task step is the target of a jump, the jump must be deleted before the step can be disabled.

Procedure

1. In the **Task Editor** grid, click the listing for the task you want to modify.



Tip:

If the grid interface is not displayed, click the **Task** button. If you do not see the task listing in the grid, expand the group that the task belongs to by clicking the downward-pointing arrow beside the group listing.

2. Click the **Edit** button.

The editing interface appears in the **Task Editor** workspace.

3. Click the **Details** tab, and then select the listing for the task step that you want to disable.

4. In the **Status** area, click **Disable**.

5. Click **Save**.

The step is disabled.

Disable a task schedule

If a schedule is no longer applicable to a task, you can disable the schedule. If the schedule is the only one configured for the task, disabling the schedule deactivates the task. You can see schedule activation state under the schedule name on the **Schedules** tab.

Procedure

1. In the **Task Editor** grid, click the listing for the task you want to modify.



Tip:

If the grid interface is not displayed, click the **Task** button. If you do not see the task listing in the grid, expand the group that the task belongs to by clicking the downward-pointing arrow beside the group listing.

2. Click the **Edit** button.

The editing interface appears in the **Task Editor** workspace.

3. Click the **Schedules** tab.

4. Select the listing for the schedule that you want to disable.

5. In the **Status** area, click **Disable**.

6. Click **Save**.

The schedule is disabled.



Note:

If the schedule is the only one configured for the task, the task is deactivated.

Deactivate a task

If a procedure is no longer applicable, you can deactivate the corresponding eSOP task so that it is unavailable during run time. You deactivate a task by disabling all related schedules.

Procedure

1. In the **Task Editor** grid, click the listing for the task you want to modify.



Tip:

If the grid interface is not displayed, click the **Task** button. If you do not see the task listing in the grid, expand the group that the task belongs to by clicking the downward-pointing arrow beside the group listing.

2. Click the **Edit** button.

The editing interface appears in the **Task Editor** workspace.

3. Click the **Schedules** tab.
4. Click the **Disable All** button.
5. Click **Save**.

All schedules related to the task are disabled, and the task is deactivated.

Use keywords in eSOP task text

For any field containing text that may be displayed in the Task List, as well as for the subject and body of task-related email messages, you can type *Keywords* as part of your entry. At run time, keywords are replaced in the Task List with parameter values specific to the given task or task step, such as the name of the user who is performing the current activity.

Procedure

1. In the **Task Editor**, click the task listing, and then click **Keywords** tab.

A read-only grid appears, displaying a list of keywords configured for the workflow template that the task is associated with.

2. For each keyword that you want to use, make note of the corresponding name.
3. If you want to use the keywords in relation to a task step, select the listing for the step.

4. Click the tab containing the text field where you want to include keywords as part of your entry, and then click in the field itself.
5. In the text location where you want the parameter value for the keyword to appear, enter the keyword, using the convention **<@Keyword@>**.

Chapter 12. ActiveX Task Controls

Home

ActiveX Task Controls

Task Controls in GE HMIs

The Workflow Task List can be integrated into other GE HMI applications.

Workflow task controls can be integrated into existing HMI applications, such as HMI/SCADA CIMPLICITY and HMI/SCADA iFIX. Workflow tasks are displayed in your application so you can operate them from within these applications. You use the ActiveX Task List and Task Indicator controls to interact with and view workflow tasks. The ActiveX Task List provides users with a list of tasks for a workflow. The Task Indicator provides a user with relevant information regarding the status of the ActiveX Task List.

Install the ActiveX Task List

Before you begin



Important:

Product Options Installation Information. When using the ActiveX Task List, product option updates made on the server are not automatically deployed to applications using the Task List. The `ProficiencyUpdater.exe` application must be run manually, with administrative privileges, in order to deploy production option updates on each client machine using ActiveX Task List.



Note:

At a single workstation, only one Task Indicator object and one ActiveX Task List object can be connected at a time per HMI application instance.

Procedure

1. From the application splash screen, click **Install ActiveX Task List**.



Note:

If the splash screen does not appear, double-click `SetupTaskList.exe` from the root directory of the installation folders.

The **License Agreement** page appears.

2. Review the license agreement, and then click **I Agree**.

The **Installation Folder** page appears.

3. Accept the default destination folder or browse for a new location, and then click **Next**.

The **Proficy Workflow Server Configuration** page appears.

4. In the **Server Name** field, enter the name of the server that you want to install the ActiveX Task List on.

The **Security Certificates** page appears.

5. The security certificates are automatically downloaded from the server you are connecting to.
6. Click **Next**.

The **Configure Task List Authentication** page appears.

7. From the **Authentication Type** drop-down menu, select one of three login options, and then click **Next**.

If you want the user name and password:

- entered manually, select **Manual Authentication**, or
- exposed in plain text, select **Control Property Authentication**, or
- saved automatically after initial setup, select **Password Encryption Authentication**.



Note:

You must first configure users using the **Configure Client** tool.

The **Installation Confirmation** page appears.



Note:

At any time during the configuration process, you can click **Back** to return to a previous page to change your settings.

8. Click **Install**.

The Installation page appears, displaying the status of each installation step.

9. When the installation is complete, click **Exit**.

Modify the ActiveX Task List installation configurations

Before you begin



Note:

To change the server name, use the Configure Client tool. For more information, see **Modify the server configuration for a remote client**. If you install the ActiveX Task List on a different machine than the application server, you must access the Configure Client tool from the following location: `../Program Files/Proficy/Proficy Task List/Program`.

Procedure

1. From the **Start** menu, select the **Task List Configuration**.

The **Proficy Task List Configuration** page appears.

2. To add or modify a server, in the **Server Name** section, enter the fully qualified name of a server.
3. To modify the login authentication, in the **Server Authentication** section:

Select...	To...
Manual Login	prompt the user to enter a user name and password at each login.
Control Property	allow the user to view the user name and password in the host's object properties.
Password Encryption	log the user in automatically.

4. If you selected **Password Encryption** in the **Server Authentication** section, add user credentials in the **User Configuration** section:
 - a. In the **Users** panel, click **Add**.
 - b. In the **Details** panel, in the **User Name** field, enter an existing Workflow user name.
 - c. In the **Password** field, enter your user account password.
 - d. In the **Confirm Password** field, enter your password again.
5. Click **Save**.

Task List Configuration

The configuration tool allows you to modify server information and its authentication credentials.

Before configuring the ActiveX Task List, note the following information:

- Each installation of the ActiveX Task List can connect to only one Workflow server at a time.
- At a single workstation, only one Task Indicator object and one ActiveX Task List object can be connected at a time per HMI application instance.
- Workflow task controls and GE HMIs are related, but work independently from each other. Therefore, user names and logins are separate.
- Like the Workflow client, the NET. TCP Port Sharing service must be enabled to use the Task List in a host application.
- To change the server name, use the Configure Client tool. For more information, see **Modify the server configuration for a remote client**. If you install the ActiveX Task List on a different machine than the application server, you must access the Configure Client tool from the following location:
`../Program Files/Proficy/Proficy Task List/Program.`

Configure a display in a host application

Before you begin



Note:

At a single workstation, only one Task Indicator object and one ActiveX Task List object can be connected at a time per HMI application instance.

Procedure

1. In your host application, add an OLE object. For more information, see the host application's online help.
2. To create a control, select:
 - **Proficy Task List Control**, or
 - **Proficy Task List Indicator Control**.
3. To configure the control's properties, right-click the control, and then:

Select...	To enter or modify...
EquipmentName	the name for a linked piece of equipment to be used in the Task List. <div data-bbox="873 1066 922 1119" data-label="Image"> </div> <div data-bbox="938 1079 1015 1110" data-label="Section-Header"> <h4>Note:</h4> </div> <div data-bbox="938 1123 1404 1291" data-label="Text"> <p>This name must match the S95 ID for the equipment found in the Universal Browser. For more information, see <i>Add a resource to the equipment model</i>.</p> </div>
ProficySOALoginUser	a login user name. <div data-bbox="873 1423 922 1476" data-label="Image"> </div> <div data-bbox="938 1436 998 1467" data-label="Section-Header"> <h4>Tip:</h4> </div> <div data-bbox="938 1480 1388 1564" data-label="Text"> <p>Property configuration is not required for Manual Login.</p> </div>
ProficySOALoginPassword	a login password.

4. Configure your host application to handle selected events.

Related reference

[Task List in HMIs \(on page 1694\)](#)

Related information

[Create optional displays and events in a host application \(on page 1693\)](#)

Create optional displays and events in a host application

Procedure

1. In your host application, select an existing OLE object. For more information, see the host application's online help.
2. For server connection, call the following methods:



Tip:

If you want operator interaction to call methods, you can add a button or create an event to call these methods.

- a. For the Task List Control, select:
 - **ConnectTaskList**, or
 - **DisconnectTaskList**
 - b. For the Task List Indicator Control, select one of the following options:
 - **ConnectTaskListIndicator**
 - **DisconnectTaskListIndicator**
3. Using code, you can do the following:
 - add a task event
 - execute code when an event occurs
 - associate controls with a run-time message
 - change a piece of equipment that is linked in the ActiveX Task List

User Configuration

When your system requires password encryption for server authentication, you must add user names and passwords. This section is enabled only when the Password Encryption option in the Server Authentication section is selected.



Note:

For the server authentication to be successful, users must have already been configured in Workflow.

Server Authentication

There are three different methods for selecting how a user connects to the server in order to access the Task List.

Manual Login

Select this option to prompt the user to manually enter a user name and password when accessing the Task List.

Control Properties

Select this option to require the user to configure the user name and password through the control properties of the HMI host.

Attribute	Value	Description
ProficySOALoginUser	User-defined	Indicates the user's login name.
ProficySOALoginPassword	User-defined	Indicates the user's login password.

Password Encryption

If you want the user names and passwords to be preconfigured, select this option. These credentials are added in the User Configuration section prior to accessing the ActiveX Task List. In the object's properties, the password is encrypted.

Task List in HMIs

The Task List is an ActiveX control that displays the Workflow Task List in a host application. The ActiveX Task List provides users with a list of tasks for a workflow.

For users to be able to see the ActiveX Task List in a host application, security permissions must be assigned to that user. These permissions are configured in Workflow. For more information, see *Understanding Security and Key Sets*.



Note:

At a single workstation, only one Task Indicator object and one ActiveX Task List object can be connected at a time per HMI application instance.

**Important:**

For iFIX users:

To enable tabbing within the ActiveX Task List, you must set the `IsSelectable` property to **True** on the ActiveX Task List, and when it first loads, you must manually select it with your mouse so that it is selected.

Control Properties for ActiveX Task List

When configuring the ActiveX Task List control, you can change the equipment context, which provides more flexible equipment options during run time. The ActiveX Task List is then filtered to the specified equipment. You can also change user login information used in the ActiveX Task List.

Property	Value	Description
Account Type	UNP	Changes the values of normal Proficiency user login credentials.
Account Type	SSO	Users can log in using their SSO credentials.
EquipmentName	N/A	S95ID for the piece of equipment added to the equipment model that is linked in the ActiveX Task List. If you have multiple pieces of equipment with the same name, only the first piece of equipment matching that name is returned.



Note:
You cannot use Equipment Classes for equipment changes.

Methods for ActiveX Task List

There are two methods available for server connection.

- ConnectTaskList
- DisconnectTaskList

Login information can change during run time when the user who is logged into the server changes. The ActiveX Task List displays after authentication has been performed.

If you receive a connection error message, review the following document for more information: *Running with a Standard Windows User Account*.



Note:

The ActiveX Task List and Task Indicator share a connection; therefore, if one is connected, the other uses that connection.

Events for ActiveX Task List

Task events can be configured after installing the ActiveX Task List to provide more information regarding the status of the ActiveX Task List. Task event information can be displayed to users if a feedback message event is part of the event's configuration.

All the task level events expose the task ID and display name with the event. The ActiveX Task List registers to the Workflow TaskInstanceChangedEvent for task level events.



Note:

For the Mobile-sized Task List, the only two events available are NewTask and TaskExpired.

Property	Parameters	Description
NewTask	string taskId, string taskName	Starts a new task. Return type: void.
TaskExpired	string taskId, string taskName	Event fires when a task that is already in your ActiveX Task List expires. It also fires when a new task is added to your list that is already expired, or has task steps assigned to you that are already expired. Return type: void.
NewFeedbackMessage	string messageType, string messageString	Displays feedback messages to users. The event is triggered

Property	Parameters	Description
		when there is a new message. Return type: void.

Task Indicator in HMIs

The Task Indicator provides a user with relevant information regarding the status of the ActiveX Task List.

The ActiveX Task List and the Task Indicator work together. When the ActiveX Task List is connected, the Task Indicator does not display notifications; however, it changes color if the status of the ActiveX Task List changes.

The indicator can be configured to integrate visually with the host application by changing the size and background color of the indicator.



Note:

At a single workstation, only one Task Indicator object and one ActiveX Task List object can be connected at a time per HMI application instance.

Control Properties for Task Indicator

When configuring the ActiveX Task List control, you can change the equipment used in the workflow. The ability to change the equipment context provides more flexible equipment options during run time.

Property	Value	Description
BackgroundColor	User-defined	Changes the color of the control.

Format the Task Indicator in a host application

Procedure

1. In your host application, modify an OLE object. For more information, see the host application's online help.
2. To associate this control with a run-time message, edit the script.
3. To allow a user to connect directly to the ActiveX Task List from the Task Indicator, select the **TaskListIndicatorClick** event and edit the script.

**Note:**

A user must already have entered authentication and be logged in.

4. **Optional:** To configure the format settings (such as size or background color), go to the object's properties.

Methods for Task Indicator

There are two methods available for configuring server connection.

- ConnectTaskListIndicator
- DisconnectTaskListIndicator

**Note:**

The ActiveX Task List and Task Indicator share a connection; therefore, if one is connected, the other uses that connection.

Events for Task Indicator

Task events can be configured after installing the ActiveX Task List to provide more information regarding the status of the Task Indicator. Task event information can be displayed to users if a feedback message event is part of the event's configuration.

Task Events

All the task level events expose the task ID and display name with the event. The Task Indicator registers to the Workflow **TaskInstanceChangedEvent** for task level events.

Property	Parameters	Description
NewTask	string taskId, string taskName	Starts a new task. Return type: void.
TaskExpired	string taskId, string taskName	Event fires when a task that is already in your ActiveX Task List expires. It also fires when a new task is added to your list that is already expired, or has task steps

Property	Parameters	Description
		assigned to you that are already expired. Return type: void.
NewFeedbackMessage	string messageType, string messageString	Displays feedback messages to users. The event is triggered when there is a new message. Return type: void.



Important:

Validation errors are only raised through the **NewFeedbackMessage** event on the ActiveX Task List. All other feedback messages related to the Task Indicator, such as a task instance termination message, can be displayed through the Task Indicator.

Other Task Events

Property	Value	Description
TaskListIndicatorClick	User-defined	Triggered when a user clicks the connected and active Task Indicator. Return type: void.
ConnectionLost	User-defined	Triggered when the server connection is lost. Return type: void.
ConnectionRestored	User-defined	Triggered when the server connection is restored. Return type: void.
ConnectResult	User-defined	Triggered when the server connection finishes and the result is passed back as an event parameter. Return type: void.

Server State Icons for Task Indicator

Indicator icons provide visual information about the status of the server connection, as well as the status of tasks. If configured to do so, a status message can also be displayed to provide more information to the user.

The following table describes the icons associated with the Task Indicator:

Icon	Status	The indicator is...
	not connected	static when it is initially loaded or when a connection is lost.
	connecting	green and animated when it is in the process of connecting to the server.
	connected	enabled when a connection succeeds.
	task waiting	flashing when new tasks to be performed are available.
	task expired	flashing when an available task expires.
	lost connection	yellow and animated when a previously connected server loses connection.

Chapter 13. Service Interfaces

Service Interfaces

HTTPS Web Services

Workflow publishes secure web service endpoints for system, workflow, and custom service providers. These endpoints use TLS/SSL security with user name and password authentication.



Note:

Administrators are unable to log in using the web service interface.

Endpoint Addresses

For web services, endpoints get published to a specific address.

```
https://<ServerName>:<Port>/< ServerInstanceName >/<NamespaceName>/  
<ContainerName>/<ServiceDisplayName>/<InterfaceName>
```

The preferred endpoints use HTTPS Port 1, which is configured to 8202 by default. To access these endpoints, configure TLS/SSL and provide a user name and password with each request.

Field	Property	Description
ServerName	User-defined	Specifies the name of the computer that has the server running the main application server.
Port	User-defined	Specifies the port used to access the web service. This default port is 8202. However, this port can be changed using the Configure Server tool at Advanced Configuration > Service Ports > HTTPS Port 1 .
ServerInstanceName	User-defined	Specifies the server name. The default name is SOAServer. However, this name can be changed using the Configure Server tool.

Field	Property	Description
NamespaceName	System-defined	Specifies the custom service provider, which is part of the service provider's product option file. The namespace is configured when this product option file is created using the Proficy Packager tool.
ContainerName	System-defined	Specifies the custom service provider, which is part of the service provider's product option file. The container name is configured when this product option file is created using the Proficy Packager tool.
ServiceDisplayName	System-defined	Specifies the custom service provider, which is part of the service provider's product option file. The service display name is defined in the service provider source code.
InterfaceName	System-defined	Specifies the custom service provider, which is part of the service provider's product option file. The interface name is the service contract name defined in the service provider source code.

Metadata Endpoint

Each service also provides a metadata endpoint for accessing service metadata, and the WSDL file can be accessed using that endpoint. The metadata endpoint address is on the first interface when the interfaces are in alphabetical order by the full interface name. For system interfaces, this is typically `IBrowse`.

For example, to access the `IWorkflowWebService` WSDL file, use the following address:

<https://<ServerName>:8202/SOAServer/SOAPProject/Workflow/Workflow/IBrowse?wsdl>

SSL Security Configuration

Each Workflow server instance must have a unique certificate for SSL security, which is either a trusted certificate obtained from a trusted root certificate authority or a self-signed certificate. The certificate's common name must be set to the name of the server machine used in URLs.



Note:

Self-signed certificates are less secure than trusted certificates and require additional configuration on each client.

For more information, refer to Microsoft® documentation for instructions on obtaining certificates and SSL security configuration.

User Permissions Between a Workflow and Service Provider

You can invoke a security check when service provider method calls are made within a workflow

Security Using Call Methods

When configuring a Call Method activity, go to the **Security** area to ensure that, at run time, the current user who has acquired the task step has permission to call the method.

Two pieces of information are required to secure a call between a workflow and a service provider. These two assessments ensure that user permissions are enforced before a method is called.

Caller context	Evaluates the user against the target and determines if the correct permissions are in place.
Method information	Determines whether the method requires a permission check.



Note:

For this feature to function correctly, you must configure a subprocess in one of the following ways to ensure that a user acquires the task step. Then, to verify that the task step is acquired, check the Current User property on the subprocess to ensure it is not null.



- Manual Start is set to `True`.
- A Form activity is used before the method call.

User Interface

Property	Value	Description
Enforce User Permissions	N/A	Select this check box to ensure that the user performing a task at run time has permission to call the method.

Enforce user permissions when invoking a service method in a workflow

You can configure a Call Method activity to ensure user permissions are checked when a service provider method is called.

Procedure

1. In the **Security** area, select the **Enforce User Permissions** check box.
2. Click **Save**.
3. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

Workflow Methods

Workflow Methods

Workflow provides a set of core service interfaces and call methods for you to use when creating a workflow. This section describes these core service interfaces and the associated call methods.

Configure Parameters for a Call Method

You can configure parameters for a selected call method.

About this task

Use this procedure to configure parameters for any call method.

Procedure

1. In the navigator, select **Workflow > Workflows**, and then create a new Workflow or select a Workflow from the list.
2. From the **Displays** panel, select **Workflow Management > Workflow Editor**.
The **Workflow Editor** page appears.
3. Select the **Workflow Editor** tab.
4. From the process map, choose from the following:
 - Select **Local/Global Subprocess** from the **Standard Activities** menu, and then drag it onto the Workflow diagram to create a new subprocess.
 - Double-click an existing **Subprocess** to display it in the editor.
5. From the subprocess map, choose from the following:
 - Select **Call Method** from the **Standard Activities** menu, and then drag it onto the Subprocess diagram to create a new call method.
 - Select an existing **Call Method** to display it in the editor.

The parameters for the call method appear in the configuration panel.
6. Click **Configure Method** in the configuration panel.
The **Configure Method** dialog box appears.
7. From the **Navigate** tab, select a call method from the list.



Note:

You can select **Next** to display the parameters configuration panel for the selected call method.

8. Click **Finish**.
The selected call method appears below **Method** in the configuration panel.
9. Enter or edit values for each parameter in the selected call method, and then click **Save**.
10. Click **Run** () to run the Workflow and the configured call method.

IArchiveManagement

IArchiveManagement allows you to create an archive file and select the data to include.

On the activity configuration panel, you can use a bind or expression to define your input and output parameters.

Subtopics in this section describe each service method for this service interface.

ArchiveData (IArchiveManagement)

Specifies the unused runtime data that you want to archive and creates an archive record when the call method runs.

Inputs	Data Type	Description
ArchiveDescription	String	Specifies a description for the archive file.
ArchiveEndTime	DateTime	Specifies the age limit for the data to be included in the archive file. That is, only unused data that is older than the specified end time will be included in the archive.
ArchiveName	String	Specifies a display name for the archive file.
Events	Boolean	Indicates whether historical event occurrence data should be archived. The default is True.
Material	Boolean	Indicates whether closed material lots and their child sublots should be archived. The default is True.
Production	Boolean	Indicates whether closed work requests and related data should be archived. The default is True.
Regulatory Compliance	Boolean	Indicates whether audit and electronic signature records should be archived. The default is True.
Resource Links	Boolean	Indicates whether user-created references should be archived. The default is True.

Inputs	Data Type	Description
Workflow	Boolean	Indicates whether workflow tracking data should be archived. The default is True.

Outputs	Description
ArchiveResponse	Returns the archive result, including whether the archive failed or succeeded, and errors and messages.

IAuditTrail

IAuditTrail allows you to write entries to the audit trail.

On the activity configuration panel, you can use a bind or expression to define your input and output parameters.

Subtopics in this section describe each service method for this service interface.

WriteAuditEntry (IAuditTrail)

Creates the entry of a specific topic written to the audit trail.

Inputs	Data Type	Description
Context	String	Specifies the contextual information that relates the entry to relevant data.
ESignature	ESignatureId	Specifies the electronic signature associated with the audited operation or event.
Location	String	Specifies the location associated with the audited operation or event.
Message	String	Specifies the message to include in the audit entry.

Inputs	Data Type	Description
Occurrence Time	DateTime	<p>Specifies the date and time that the action being audited occurred. This entry can be null.</p> <p>In the Inputs section of the activity configuration panel, click the drop-down arrow to set the date and time.</p>
User	String	Specifies the user associated with the audited operation or event.

No data is returned.

IDatabaseService

IDatabaseService provides access to relational data, such as databases.

On the activity configuration panel, you can use a bind or expression to define your input and output parameters.

Subtopics in this section describe each service method for this service interface.

ExecuteProcedure (IDatabaseService)

Executes a stored procedure against a database connection, and provides for returning results in either the actual format(s) of the specified parameter(s) or in string format.

In the **Configure Method** dialog box, click **Next** to select the connection and procedure. Next, specify whether you want results returned in the actual format(s) of the parameter(s) in question (**Return typed values**, the default) or string format (**Return values as strings**), and then click **Finish**. Your connection and procedure selections determine the specific input parameters for the method, which you configure in the **Inputs** section of the activity configuration panel.



Note:

Beginning with release 2.2 SP1, this method includes a new parameter (ConvertValues) that allows you to specify whether returned values should be automatically converted to string format,



as was done in past releases. Prior implementations of this method will continue to function as in the past.

Inputs	Data Type	Description
Connection	DirectoryResource	Specifies the database connection that is used to execute a particular stored procedure.
ProcedureName	String	Specifies the name identified for a specific procedure.
ConvertValues	Boolean	<p>Specifies whether to convert returned values to string format. Options are as follows:</p> <ul style="list-style-type: none"> • False, to leave returned values in their native data formats (the default). <div data-bbox="1105 1003 1419 1314" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note: In the Configure Method dialog box, this option equates with Return typed values.</p> </div> <ul style="list-style-type: none"> • True, to convert returned values from their native data formats to string format. <div data-bbox="1105 1549 1419 1745" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note: In the Configure Method dialog box, this option</p> </div>

Inputs	Data Type	Description
		 equates with Return values as strings .
Inputs	List<Parameter>	Specifies a collection of input parameter values and output parameter initial values that can be passed to a database connection to execute a stored procedure.

Outputs	Data Type	Description
ReturnValue	StoredProcedureExecutionResult	The execution result, including execution time, collection of output parameter values, and collection of DataTable (Result), of the stored procedure against a particular database.  Note: Data transfer limit: 10 MB.

ExecuteStatement (IDatabaseService)

Executes a query or update statement against a database connection, and provides for returning results in either the actual format(s) of the specified parameter(s) or in string format.

In the **Configure Method** dialog box, click **Next** to select whether you want to use an existing statement or configure a new statement. In the latter case, clicking the corresponding **Edit** link opens a dialog box, where you can specify the statement type and provider and then configure the statement itself. Next, specify whether you want results returned in the actual format(s) of the parameter(s) in question (**Return typed values**, the default) or string format (**Return values as strings**), and then click **Finish**.

**Note:**

Beginning with release 2.2 SP1, this method includes a new parameter (ConvertValues) that allows you to specify whether returned values should be automatically converted to string format, as was done in past releases. Prior implementations of this method will continue to function as in the past.

Inputs	Data Type	Description
Connection	DirectoryResources	Specifies the database connection that is used to execute a particular statement.
Statement	StatementInfo	Specifies the statement information to execute: <ul style="list-style-type: none"> • Statement (select, update, insert, delete) • Statement Type (query, update) • Provider (SQL, ODBC)
ConvertValues	Boolean	Specifies whether to convert returned values to string format. Options are as follows: <ul style="list-style-type: none"> • False, to leave returned values in their native data formats (the default). <div data-bbox="1105 1419 1419 1734" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin: 10px 0;">  <p>Note: In the Configure Method dialog box, this option equates with Return typed values.</p> </div> <ul style="list-style-type: none"> • True, to convert returned values from their native

Inputs	Data Type	Description
		<p>data formats to string format.</p> <div data-bbox="1105 380 1421 741" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;"> <p> Note: In the Configure Method dialog box, this option equates with Return values as strings.</p> </div>

Outputs	Data Type	Description
ReturnValue	StatementExecutionResult	<p>The execution result, including execution time, affected rows, and collection of DataTable (Result), of the statement against a particular database. Affected rows are provided based on statement type, particularly when StatementType is Update.</p> <div data-bbox="1024 1262 1421 1440" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;"> <p> Note: Data transfer limit: 10 MB.</p> </div>

IDirectorySearch

IDirectorySearch allows you to search the Directory of Resources for a set of resources based on a search query, or for a list of resources based on a resource class.

On the activity configuration panel, you can use a bind or expression to define your input and output parameters.

Subtopics in this section describe each service method for this service interface.

CreateClassQueryExpression (IDirectorySearch)

Creates a query expression to return resources associated with a given class.

Inputs	Data Type	Description
Class Resource	DirectoryResource	Specifies the resource class you want to use for your query expression. In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.

Outputs	Description
ClassificationExpression	Returns a list of all resources associated with the selected resource class.



Important:

The returned collection of data is based on user-defined values and is dependent on your published data. For example, if you select the personnel class "Operator" for your query expression, then a list of all personnel resources associated with that class is returned.

Search (IDirectorySearch)

Returns an array of directory resources for a given search string.

This method provides the ability to filter results to include resources with names or descriptions that contain but do not exactly match the specified filter text. For exact-match filtering, use the *SearchEx (IDirectorySearch)* call method instead.

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

**Note:**

To test your search strings, click **Next** in the *Configure Method* dialog box. Any values entered here are used as values in the Inputs section of the activity configuration panel. For more information, see *Advanced Search*.

Inputs	Data Type	Description
DescriptionFilter	String	<p>Specifies text contained in resource descriptions. Resource descriptions containing this text are included in the search results. If you leave this parameter empty, description filtering is NOT done.</p>
ExcludeClause	String	<p>Specifies the classifications that you want to exclude from the search.</p> <p>In the Configure Method dialog box, click Next, and then click Advanced Search. Finally, from the Classifications pane, select the applicable option, and then click Add under Excluded Classifications area.</p> <p>Valid values include any classification available in the Directory of Resources, such as:</p> <ul style="list-style-type: none"> • Equipment Resource • Type - Enterprise • Type - Work Request • Category - <category name>

Inputs	Data Type	Description
		 Note: This is not a comprehensive list. If you want to exclude just a single Class classification, you can configure a <i>Create-ClassQueryExpression</i> call method and bind this parameter to the output of that method instead.
IncludeClause	String	<p>Specifies the classifications that you want to include in the search.</p> <p>In the Configure Method dialog box, click Next, and then click Advanced Search. Finally, from the Classifications pane, select the applicable option, and then click Add under Included Classifications area.</p> <p>Valid values include any classification available in the Directory of Resources, such as:</p> <ul style="list-style-type: none"> • Equipment Resource • Type - Enterprise • Type - Work Request • Category - <category name>  Note: This is not a comprehensive list. If you want to exclude just a single

Inputs	Data Type	Description
		 Class classification, you can configure a <i>Create-ClassQueryExpression</i> call method and bind this parameter to the output of that method instead.
NameFilter	String	Specifies text contained in resource names. Resource names containing this text are included in the search results. If you leave this parameter empty, name filtering is NOT done.
StartingResource	String	Specifies the model to start your search at. The default is <i>Publications</i> . Valid values are any models that appear in the Models list in the navigator, such as <i>Production</i> , <i>Personnel</i> , <i>Workflow</i> , and so on. In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.

Outputs	Description	
DirectoryResource[]		Returns all of the resources that match the defined search criteria. Results are limited to 5,000. A search returning more than the limit generates an exception.

Outputs	Description	
		Use a <i>For Each Activity</i> to loop through all resources or use an <i>Index Activity</i> with an integer value. Indexing is supported only by the integer value; it does not support string values.

Convert a resource's display name to a Directory Resource

You can convert a resource's String data type display name to a Directory Resource data type using the IDirectorySearch Search call method.

Before you begin



Note:

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with



Note:

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Drag and drop a **Local Subprocess** into the workflow and double-click it, or double-click an existing subprocess.



Note:

If required, configure the workflow.

6. Drag and drop a **Call Method** into the subprocess and double-click it, or double-click an existing Call Method activity.
7. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.

8. Double-click **IDirectorySearch** to display the list of available call methods.
9. Click **Search**, and then do one of the following:

Click...	To configure...
Finish	basic search results.
Next	advanced search results using classification restrictions, and then click Finish .

The input parameters appear in the activity configuration panel.

10. In the **Name Filter** field, from the drop-down arrow:

Click...	To...
Bind	bind the property to the output of a previous activity.
Expression	insert an expression.

11. In the **Starting Resource** field:

Click...	To select a...
Bind	node whose children and descendants you want to search.
	<div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Tip: Use the bind path that is as close to the starting resource as possible. For example, if you want to find a Directory Resource for a production segment, select the Production Segments folder in the Production model. The closer the folder is to the actual resource, the better the search performance.</p> </div>
Ellipsis button	select a resource to search from the Universal Browser .

12. To return multiple search results:

- Search a resource with a similar Display Name.

**Note:**

You can return partial matches. For example, if you want to find a resource named **MyEquipment** and you search for **My**, all resources with **My** in the display name will be returned, including MyEquipment.

- To find the exact Display Name, insert a *For Each Activity* or a *While* loop to iterate through the results.
- To return single search results:
 - Insert an Index activity after the Call Method activity.
 - In the Collection field, from the drop-down arrow, select **Bind**. The **Select Variable** dialog box appears.
 - From the current Call Method, select the Output Parameter's **ReturnValue**, and then click **OK**.
 - In the Index field, enter 0 to return the first item in the collection. You can now access the Directory Resource through the result of the Index activity.
 - Click **Save**.
 - Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

SearchEx (IDirectorySearch)

Returns an array of directory resources for a given search string, with the ability to filter results based on whether the value specified for the resource name or description parameter partially or exactly matches the actual value for the given parameter.

**Note:**

To test your search strings, click **Next** in the **Configure Method** dialog box. Any values entered here are used as values in the Inputs section of the activity configuration panel. For more information, see *Advanced Search*.

Inputs	Data Type	Description
Description Filter	String	Specifies text contained in resource descriptions. Resource descriptions containing this text are included in the search results. If you leave this parameter emp-

Inputs	Data Type	Description
		ty, description filtering is NOT done.
Exclude Clause	String	<p>Specifies the classifications that you want to exclude from the search.</p> <p>In the Configure Method dialog box, click Next, and then click Advanced Search. Finally, from the Classifications pane, select the applicable option, and then click Add under Excluded Classifications area.</p> <p>Valid values include any classification available in the Directory of Resources, such as:</p> <ul style="list-style-type: none"> • Equipment Resource • Type - Enterprise • Type - Work Request • Category - ,category name> <div data-bbox="1024 1304 1425 1799" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note:</p> <p>This is not a comprehensive list. If you want to exclude just a single <i>Class</i> classification, you can configure a <i>Create-ClassQueryExpression</i> call method and bind this parameter to the output of that method instead.</p> </div>

Inputs	Data Type	Description
Filter Matching Option	Search Filter Options	<p>Specifies the algorithm applied to match the <i>Name Filter</i> and <i>Description Filter</i> text to candidate resources found by the search.</p> <p>Options are:</p> <ul style="list-style-type: none"> • PartialMatchCaseInsensitive: Includes any candidate resources with names and descriptions that contain the filter text regardless of text case. This is the default setting. • ExactMatchCaseInsensitive: Includes any candidate resources with names and descriptions that exactly match the filter text regardless of text case <p>In the Inputs section of the activity configuration panel, click the ellipsis button, and then select the option you want to use.</p>
Include Clause	String	<p>Specifies the classifications that you want to include in the search.</p> <p>In the Configure Method dialog box, click Next, and then click Advanced Search. Finally, from the Classifications pane, select the applicable option, and then click Add under Included Classifications area.</p>

Inputs	Data Type	Description
		<p>Valid values include any classification available in the Directory of Resources, such as:</p> <ul style="list-style-type: none"> • Equipment Resource • Type - Enterprise • Type - Work Request • Category - ,category name> <div data-bbox="1027 695 1419 1184" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: This is not a comprehensive list. If you want to exclude just a single <i>Class</i> classification, you can configure a <i>Create-ClassQueryExpression</i> call method and bind this parameter to the output of that method instead.</p> </div>
Name Filter	String	Specifies text contained in resource names. Resource names containing this text are included in the search results. If you leave this parameter empty, name filtering is NOT done.
Starting Resource	String	<p>Specifies the model to start your search at. The default is <i>Publications</i>.</p> <p>Valid values are any models that appear in the Models list in the navigator, such as <i>Production</i>, <i>Personnel</i>, <i>Workflow</i>, and so on.</p>

Inputs	Data Type	Description
		In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.

Outputs	Description	
Directory Resource[]		Returns all of the resources that match the defined search criteria. Results are limited to 5,000. A search returning more than the limit generates an exception. Use a <i>For Each Activity</i> to loop through all resources or use an <i>Index Activity</i> with an integer value. Indexing is supported only by the integer value; it does not support string values.

IService

You can use IService to send emails as part of a workflow process.

On the activity configuration panel, you can use a bind or expression to define your input and output parameters.

Subtopics in this section describe each service method for this service interface.

SendEmail (IService)

You can add email addresses to a workflow to send an email message to several recipients. These emails are sent from the default sender's preconfigured email address.

Input Data



Note:

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Inputs	Data Type	Description
Recipients	IEnumerable[[DirectoryResource]]	Lists the person or group from the Personnel model who will receive the email. All recipient email addresses must be configured in order to avoid errors. Click the drop-down arrow to bind email recipients to the return value.
Subject	String	Specifies the subject heading of the email being sent.
Body	String	Specifies the body of the email being sent.

SendEmailFromPerson (IEmailService)

You can add a single email address to a workflow to send an email message to a particular recipient from a particular sender.

Input Data



Note:

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Inputs	Data Type	Description
Recipient	DirectoryResource	Specifies the person or group from the Personnel model who will receive the email. All recipient email addresses must be

Inputs	Data Type	Description
		configured in order to avoid errors. In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.
Sender	DirectoryResource	Specifies the person or group from the Personnel model who is sending the email. In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.
Subject	String	Specifies the subject heading of the email being sent.
Body	String	Specifies the body of the email being sent.

SendEmailSingle (IEmailService)

You can add an email address to a workflow to send an email message to a single recipient. These emails are sent from the default sender's preconfigured email address.

Input Data



Note:

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Inputs	Data Type	Description
Recipient	DirectoryResource	Specifies the person or group from the Personnel model who

Inputs	Data Type	Description
		will receive the email. All recipient email addresses must be configured in order to avoid errors. In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.
Subject	String	Specifies the subject heading of the email being sent.
Body	String	Specifies the body of the email being sent.

SendEmailWithAttachments (IEmailService)

You can send an email message from the Workflow server with optional attachments to multiple recipients.

Make sure that the email configuration for the server has been set up.

When configuring Workflow emails, recipients can be specified using the parameters for `Recipient Emails`, `Recipient Resources`, or both. At least one recipient must be specified.

1. Click **Configure Method** to select a call method. See **Configure Parameters for a Call Method** for more information.
2. Enter or edit values as shown in the table below, and then click **Save**.
3. Click **Run** () to run Workflow and the configured call method.

Inputs	Data Type	Description
Recipient Emails	String[]	Email message recipients are expressed as email addresses. In the Inputs section of the configuration pane, click the ellipsis (...) to open the String Collection Editor dialog

Inputs	Data Type	Description
Recipient Resources	IEnumerable<DirectoryResource>	<p>box. Enter one or more email addresses into the Recipient field, and then click OK.</p> <p>Specifies a person, class, or group from the Personnel model who will receive the email. All recipients must have email addresses must be configured in order avoid errors.</p> <p>In the Inputs section of the configuration pane, click the ellipsis (...) to open the Universal Browser. From the Navigate tab, select the resource.</p>
Sender	DirectoryResource	<p>Specifies the person or group from the Personnel model who is sending the email. An email address must be configured for the sender resource.</p> <p>In the Inputs section of the configuration pane, click the ellipsis (...) to open the Universal Browser. From the Navigate tab, select the resource.</p> <div data-bbox="751 1035 1414 1167" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: If not specified, the default sender is used. </div>
Subject	String	Specifies the subject heading of the email being sent.
Body	String	Specifies the body of the email being sent.
Attachments	String	<p>In the Inputs section of the configuration pane, click the ellipsis (...) to open the String Collection Editor dialog box. Enter the file paths of attachments to include with the email message, and then click OK.</p> <div data-bbox="751 1514 1414 1688" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: Attachment file paths represent file paths on the Workflow server machine, not local paths. </div>
Enforce User Permissions	N/A	Check the Enforce User Permissions check box to use the permissions of the Workflow's current user when invoking the method.

**Tip:**

You can test the email configuration by selecting the **Test/Debug** tab and clicking **Test**.

Send emails using the email service provider

You can use the email service provider to send out emails automatically to one or more recipients during workflow run time.

Before you begin

For the email service provider to work, you must configure *permissions* by selecting the appropriate check box.

- To configure server-wide settings for sending emails, select **Configure email**.
- To allow email messages to be sent, select **Send email**.

Procedure

1. In the navigator, click **Proficy System > Proficy System**.
2. In the **Displays** panel, click **Email**.

The **Email Configuration Setup** window appears.

**Note:**

You must configure all required fields in order to avoid errors.

3. Enter the relevant data in each of the following fields.
 - a. **Required:** In the **Server host name** field, enter the server from which emails will be sent.
 - b. **Required:** In the **Server port number** field, enter the server's port number.
 - c. **Optional:** In the **Server timeout value** field, enter the number of seconds allowable for an email to be sent before the server times out.
 - d. **Required:** In the **Default sender address** field, enter the email address from which emails will be sent.
 - e. **Optional:** In the **Default sender name** field, enter a name related to the sender of the email.
 - f. **Optional:** In the **Test recipient email address** field, enter an email address to test the sender's email, and then click **Send Test Message**.
If the sender's email works, a test email arrives in the recipient's Inbox.

4. To add recipients to an email distribution list, add an *IDirectorySearch* service provider to the workflow.
 - a. Add the Search method to your workflow, following the steps in the *Convert a resource's display name to a Directory Resource* topic.
 - b. In the **Starting Resource** field, click the ellipsis button.
The **Universal Browser** appears.
 - c. From the **Personnel** folder, select the recipient that you want to send an email to, and then click **OK**. If you want to send an email to all users, select **Personnel**, and then click **OK**.
 - d. Click **Save**.
5. Drag and drop a second Call Method activity into the subprocess.
6. To send emails during workflow execution, add one of the following email service call methods:

Select...	To send an email to...
SendEmail	several recipients.
SendEmailFromPerson	a single recipient from a specific sender. To include this sender, in the Sender field, click the ellipsis button, and then select the user.
SendEmailSingle	a single recipient from the preconfigured sender.

**Note:**

Ensure that each recipient has an email address *associated* with his account.

7. In the **Recipient(s)** field, select **Bind** from the drop-down arrow.
The **Select Variable** dialog box appears.
8. From the Call Method activity that contains the Search call method, select the **ReturnValue**, and then click **OK**.
9. **Optional:** To configure other fields, from the **Inputs** section:

In this field...	Enter text to include...
Body	within the email's body section.

In this field...	Enter text to include...
Subject	in the email's subject line.

10. Click **Save**.

IEquipment

IEquipment allows you to read from and write to the Equipment model.

On the activity configuration panel, you can use a bind or expression to define your input and output parameters.

Subtopics in this section describe each service method for this service interface.

AddAspect (IEquipment)

Allows you to add an aspect to an equipment instance or class.

Inputs	Data Type	Description
Aspect Type Name	String	Specifies the name of the aspect type to add.
Equipment Instance or Class	DirectoryResource	Specifies the equipment instance or class to add the aspect to.



Note:

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Create an equipment aspect

Equipment aspects dynamically extend existing equipment instances or classes by adding new attributes and business logic without modifying or recompiling the original objects.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with.

**Note:**

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the Workflow Editor tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. Double-click **iEquipment** to display the list of available call methods.
 7. Click **AddAspect**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
 8. Configure the call method as follows:
 - a. In the **Aspect Type Name** field, enter the name of the aspect you want to add to the resource.
 - b. In the **Equipment Instance or Class** field, click the ellipsis button, and then select an instance or class to add the aspect to.
 9. Click **Save**.
 10. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

AddClass (IEquipment)

Allows you to add a class to an equipment instance.

Inputs	Data Type	Description
Equipment Class	DirectoryResource	Specifies the equipment class to add.

Inputs	Data Type	Description
Equipment Instance	DirectoryResource	Specifies the equipment instance to add the class to.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Related reference

[RemoveClass \(IEquipment\) \(on page 1757\)](#)

AddPropertyAspect (IEquipment)

Adds an aspect to an equipment instance or class property.

Inputs	Data Type	Description
aspectName	String	Specifies the property aspect to associate with the property.
Equipment Instance or Class	DirectoryResource	Specifies the equipment instance or class to add the property aspect to.
propertyName	String	Specifies the name of the property to add the aspect to.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Create an equipment property aspect

About this task

Equipment property aspects dynamically extend equipment instance or class properties without modifying or recompiling the original objects.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with.



Note:

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IEquipment** to display the list of available call methods.
 8. Click **AddPropertyAspect**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
 9. Configure the call method as follows:
 - a. In the **aspectName** field, enter the name of the property aspect you want to add to the property.
 - b. In the **Equipment Instance or Class** field, click the ellipsis button, and then select an instance or class.
 - c. In the **propertyName** field, enter the name of the property you want to add the property aspect to.
 10. Click **Save**.
 11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

AddPropertyClass (IEquipment)

Allows you to add a property of type Class to an equipment instance.

Inputs	Data Type	Description
Equipment Class	DirectoryResource	Specifies the equipment class to add as a property.
Equipment Instance	DirectoryResource	Specifies the equipment instance to add the class property to.
Property Name	String	Specifies a display name for the class property.



Note:

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

AddClassPropertyToClass (IEquipment)

Allows you to add a property of type Class to an equipment class.

Inputs	Data Type	Description
Equipment Class	DirectoryResource	Specifies the equipment class to add the class property to.
Equipment Class Property	DirectoryResource	Specifies the equipment class to add as the class property.
Property Name	String	Specifies a display name for the new class property.

AddPropertyToClass (IEquipment)

Allows you to add a property to an equipment class.

On the activity configuration panel, you can use a bind or expression to define your input and output parameters.

Inputs	Data Type	Description
Equipment Class	DirectoryResource	Specifies the equipment class to add the property to.
Property Default Value	Object	Specifies the default value of the equipment class property.
Property Description	String	Specifies the description of the class property.
Property Name	String	Specifies a display name for the class property.
Property Type	ParameterDataType	Specifies a data type for the class property.
Unif of Measure	String	Specifies a unit of measure for the class property.

CreateClass (IEquipment)

Allows you to create a new equipment class.

On the activity configuration panel, you can use a bind or expression to define your input and output parameters.

Inputs	Data Type	Description
Description	String	Specifies the description of the new equipment class.
S95 ID	String	Specifies the ID of the new equipment class.

Outputs	Description
DirectoryResource	The new equipment class.

CreateEquipment (IEquipment)

Creates a new equipment instance of a given type under the given parent.

Inputs	Data Type	Description
Description	String	Specifies the description of the new equipment instance.
Parent	DirectoryResource	Specifies the parent of the new equipment instance.
S95Type	S95Type	Specifies the type of equipment. For more information on equipment, see the <i>Equipment Model</i> .
S95 ID	String	Specifies the ID of the new equipment instance.

Outputs	Description
DirectoryResource	Address of the new equipment instance.

DeleteClass (IEquipment)

Allows you to delete an equipment class.

On the activity configuration panel, you can use a bind or expression to define your input and output parameters.

Inputs	Data Type	Description
Equipment Class	DirectoryResource	Specifies the equipment class to delete.

DeleteEquipment (IEquipment)

Deletes an equipment instance.

Inputs	Data Type	Description
Equipment Instance	DirectoryResource	Specifies the equipment instance to delete.

Inputs	Data Type	Description
		 Note: <ul style="list-style-type: none"> • If the selected equipment instance has any child resources, those child resources will also be deleted. • When a resource is deleted, it cannot be recovered.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

GetAspectTypes (IEquipment)

Returns the applicable and applied aspect types for an equipment instance or class.

Inputs	Data Type	Description
Equipment Instance or Class	DirectoryResource	Specifies the equipment instance or class to read.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description
ExtensionInfo[]	Returns an array of possible aspect information con-

Outputs	Description	
		taining aspect name, description, and a flag indicating whether the aspect is applied to the specified resource.

GetClassProperties (IEquipment)

Returns a list of properties defined by an equipment class.

Inputs	Data Type	Description
Equipment Class	DirectoryResource	<p>Specifies the address of the equipment instance.</p> <p>In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.</p>

Outputs	Description	
String[]		Returns a string array of property names.

GetDataSourceLink (IEquipment)

Gets the current data source binding for a given equipment property.

Inputs	Data Type	Description
Equipment Instance	DirectoryResource	<p>Specifies the equipment instance.</p> <p>In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs</p>

Inputs	Data Type	Description
		section of the activity configuration panel, click the ellipsis button to select a resource.
propertyName	String	Specifies the name of the property whose data source link is being retrieved.

Outputs	Description
DataAccessEndPoint	Returns a link to an external data item.

GetEquipmentChildren (IEquipment)

Returns a collection of all the child resources for the specified parent resource.



Note:

Returns the child resources one level below the specified parent.

Inputs	Data Type	Description
Equipment Instance	DirectoryResource[]	<p>Specifies the equipment resource you want to retrieve the child resources for. If the specified resource does not have any child resources, nothing is returned.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>

Outputs	Description	
DirectoryResource[]		<p>Returns a list of the user-defined child resources for the specified parent resource.</p> <p>Use a <i>For Each</i> or <i>Index</i> activity to return the resource you want.</p> <div data-bbox="1026 548 1421 722" style="border: 1px solid #0070C0; border-radius: 10px; padding: 5px;">  Note: The Index activity returns only integer values. </div>

GetEquipmentParent (IEquipment)

Returns the parent of the specified equipment resource.

Inputs	Data Type	Description
Equipment Instance	DirectoryResource	<p>Specifies the equipment resource to retrieve the parent for.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>

Outputs	Description	
DirectoryResource		<p>Returns the parent resource for the specified equipment resource. If the specified resource does not have a parent, nothing is returned.</p>

GetPropertyAspectTypes (IEquipment)

Returns the applicable and applied aspect types for an equipment instance or class property.

Inputs	Data Type	Description
Equipment Instance or Class	DirectoryResource	Specifies the equipment instance or class to read.
propertyName	String	Specifies the name of the property to read.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description
ExtensionInfo[]	Returns an array of possible aspect information containing aspect name, description, and a flag indicating whether the aspect is applied to the specified resource.

Get equipment property aspect types

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with.

**Note:**

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:

- At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IEquipment** to display the list of available call methods.
 8. Click **GetPropertyAspectTypes**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
 9. Configure the call method as follows:
 - a. In the **Equipment Instance or Class** field, click the ellipsis button, and then select an instance or class.
 - b. In the **propertyName** field, enter the name of the property to read.
 10. Click **Save**.
 11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

HasAspect (IEquipment)

Checks whether an equipment instance or class has the specified aspect applied to it.

Inputs	Data Type	Description
Aspect Type Name	String	Specifies the aspect type to check.
Equipment Instance or Class	DirectoryResource	Specifies the equipment instance or class to check.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description
Boolean	True if the selected resource is augmented by the specified aspect; otherwise, False.

Verify an equipment aspect

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with.

**Note:**

A resource must already have been added in order to select it from this list.

3. Click the **Workflow Editor** tab.
4. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
5. Double-click **IEquipment** to display the list of available call methods.
6. Click **HasAspect**, and then click **Finish**.

The input parameters appear in the activity configuration panel.
7. Configure the call method as follows:

- a. In the **Aspect Type Name** field, enter the name of the property to verify.
 - b. In the **Equipment Instance or Class** field, click the ellipsis button, and then select an instance or class.
8. Click **Save**.
 9. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

Get equipment aspect types

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with.



Note:

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
7. Double-click **IEquipment** to display the list of available call methods.
8. Click **GetAspectTypes**, and then click Finish.

The input parameters appear in the activity configuration panel.

9. In the **Equipment Instance or Class** field, select the resource from a directory click the ellipsis button, and then select the instance or class to read.
10. Click **Save**.
11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

HasClass (IEquipment)

Checks if a given instance has the given class.

Inputs	Data Type	Description
Equipment Class	DirectoryResource	Specifies the location of the equipment class.
Equipment Instance	DirectoryResource	Specifies the equipment instance to check.



Note:

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description
Boolean	True if the equipment instance is being augmented by the class; otherwise, False.

HasPropertyAspect (IEquipment)

Checks whether an equipment instance or class property has a specified aspect applied.

Inputs	Data Type	Description
attribute	String	Specifies the aspect to check.
Equipment Instance or Class	DirectoryResource	Specifies the equipment instance or class to check.

Inputs	Data Type	Description
propertyName	String	Specifies the name of the property to read.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description
Boolean	True if the resource property is augmented by the specified aspect; otherwise, False.

Verify an equipment property aspect

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with.

**Note:**

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:

- Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IEquipment** to display the list of available call methods.
 8. Click **HasPropertyAspect**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
 9. Configure the method as follows:
 - a. Enter the name of the aspect in the **aspectName** field.
 - b. In the **Equipment Instance or Class** field, click the ellipsis button, and then select the resource from Universal Browser.
 - c. Enter the name of the property in the **propertyName** field and then click **OK**.
 10. Click **Save**.
 11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

ReadClasses (IEquipment)

Returns a list of classes associated with an equipment instance.

Inputs	Data Type	Description
Equipment Instance	DirectoryResource	Specifies the equipment instance to read classes from. In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description
DirectoryResource[]	Returns the equipment classes associated with the specified instance.

ReadEquipmentClassValues (IEquipment)

Reads the values of a class property of a specific equipment instance.

Inputs	Data Type	Description
Equipment Class	DirectoryResource	Specifies the address of the equipment class defining the properties to be read.
Equipment Instance	DirectoryResource	Specifies the equipment instance to read the class values for.



Note:

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description
Equipment Values	Returns all property and attribute values for the specified class of a specific equipment instance.

ReadEquipmentValue (IEquipment)

Reads the value of a single property or attribute of a specific equipment resource.

Inputs	Data Type	Description
Equipment Instance	DirectoryResource	Specifies the equipment resource to read. In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.

Inputs	Data Type	Description
Property Name	String	<p>Specifies the property and/or attribute names to return.</p> <p>The format for a property is: propertyname.</p> <p>The format for a property and attribute is: propertyname.attribute.</p> <p>Use a comma to separate each property you want to retrieve; for example, propertyname1, propertyname2.attribute1, propertyname3.attribute1.</p>

Outputs	Description
Data Value	Returns all data about the specified property and/or attribute for the selected resource.

ReadEquipmentValues (IEquipment)

Returns the values of all of the properties and attributes of a specific equipment resource.

Inputs	Data Type	Description
Equipment Instance	DirectoryResource	<p>Specifies the equipment instance to read.</p> <p>In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.</p>

Outputs	Description	
Equipment Values		<p>Returns a list of user-defined values of the properties and attributes for the specified resource.</p> <p>To access these properties, use an <i>Index</i> activity or use a <i>ForEach</i> loop. You can use the parameter name as the index for the Index activity.</p>



Important:

The returned collection of data is based on user-defined values and is dependent on your published data. For information on the attributes that can be read from, see the related properties for the equipment model.

ReadEquipmentValuesByName (IEquipment)

Returns the values of named properties and attributes of a specific equipment resource.

Inputs	Data Type	Description
Equipment Instance	DirectoryResource	<p>Specifies the equipment instance to read.</p> <p>In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.</p>
Property Names	String[]	Specifies the names of the attributes or property names to read.

Outputs	Description	
Equipment Values		Returns a list of user-defined values of the named properties and

Outputs	Description	
		attributes for the specified resource. To access these properties, use an <i>Index Activity</i> or use a <i>For Each</i> loop. You can use the parameter name as the index for the Index activity.

**Important:**

The returned collection of data is based on user-defined values and is dependent on your published data. For information on the attributes that can be read from, see the related properties for the equipment model.

ReadPropertyAttribute (IEquipment)

Returns the value of an attribute of an equipment instance or class property.

Inputs	Data Type	Description
Attribute Name	String	Specifies the property attribute to read.
Equipment Instance or Class	DirectoryResource	Specifies the equipment instance or class to read.
Property Name	String	Specifies the name of the property whose attribute you want to read.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description	
Object		Returns the user-defined value of the attribute for the specified property.

Read an equipment property attribute

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. From the Workflows list, select the resource you want to work with.



Note:

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
7. Double-click **IEquipment** to display the list of available call methods.
8. Click **ReadPropertyAttribute**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
9. Configure the call method as follows:
 - a. In the **attribute** field, enter a name for the attribute you want to read.
 - b. In the **Equipment Instance or Class** field, click the ellipsis button, and then select the instance or class whose property you want to read.
 - c. In the **propertyName** field, enter the name of the property whose attribute you want to read.
10. Click **Save**.
11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

ReadPropertyAttributes (IEquipment)

Returns the values of all attributes of an equipment instance or class property.

Inputs	Data Type	Description
Equipment Instance or Class	DirectoryResource	Specifies the equipment instance or class to read.
Property Name	String	Specifies the name of the property whose attributes you want to read.



Note:

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description
AttributeCollection	Returns a list of the user-defined values of the attributes for the specified property.

Read multiple equipment property attributes

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. From the Workflows list, select the resource you want to work with.



Note:

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:

- At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IEquipment** to display the list of available call methods.
 8. Click **ReadPropertyAttributes**, and then click **Finish**. The input parameters appear on the activity configuration panel.
 9. Configure the call method as follows:
 - a. In the **Equipment Instance or Class** field, click the ellipsis button, and then select the instance or class whose property you want to read.
 - b. In the **propertyName** field, enter the name of the property whose attributes you want to read.
 10. Click **Save**.
 11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

ReadPropertyAttributesByName (IEquipment)

Returns the attribute names of the specified equipment instance or class property.

Inputs	Data Type	Description
Attribute Names	String[]	Specifies the names of the attributes to read.
Equipment Instance or Class	DirectoryResource	Specifies the equipment instance or class to read.
Property Name	String	Specifies the name of the property whose attributes you want to read.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description	
AttributeCollection		Returns a list of the user-defined names of the attributes for the specified property.

Read multiple specific equipment property attributes

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. From the Workflows list, select the resource you want to work with.

**Note:**

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
7. Double-click **IEquipment** to display the list of available call methods.

8. Click **ReadPropertyAttributesByName**, and then click Finish. The input parameters appear on the activity configuration panel.

The input parameters appear in the activity configuration panel.

9. Configure the call method as follows:

a. In the **attributeNames** field, click the ellipsis button to open the String Collection Editor dialog box, where you can enter the names of the attributes you want to read.



Note:

Enter only one attribute name per line.

b. In the **Equipment Instance or Class** field, click the ellipsis button, and then select the instance or class whose attributes you want to read.

c. In the **propertyName** field, enter the name of the property whose attributes you want to read.

10. Click **Save**.

11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

RemoveAspect (IEquipment)

Removes an aspect from an equipment instance or class.

Inputs	Data Type	Description
Aspect Type Name	String	Specifies the name of the aspect type to remove.
Equipment Instance or Class	DirectoryResource	Specifies the equipment instance or class to remove the aspect from.



Note:

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Remove an equipment aspect

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with.

**Note:**

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IEquipment** to display the list of available call methods.
 8. Click **RemoveAspect**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
 9. Configure the call method as follows:
 - a. In the **Aspect Type Name** field, enter a name of the aspect you want to remove.
 - b. In the **Equipment Instance or Class** field, click the ellipsis button, and then select an instance or class to remove the aspect from.
 10. Click **Save**.
 11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

RemoveClass (IEquipment)

Removes a specified equipment class from an equipment instance.

Attribute	Value	Description
Equipment Class	DirectoryResource	Specifies the equipment class to remove.
Equipment Instance	DirectoryResource	Specifies the equipment instance to remove the class from.

RemovePropertyAspect (IEquipment)

Removes an aspect from an equipment instance or class property.

Inputs	Data Type	Description
aspectName	String	Specifies the name of the property aspect type to remove.
Equipment Instance or Class	DirectoryResource	Specifies the equipment instance or class to remove the property aspect from.
propertyName	String	Specifies the name of the property to remove the aspect from.



Note:

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Remove an equipment property aspect

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with.



Note:

A resource must already have been added in order to select it from this list. For more information, see *Add a resource to the workflow model*.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.

4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IEquipment** to display the list of available call methods.
 8. Click **RemovePropertyAspect**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
 9. Configure the call method as follows:
 - a. In the **aspectName** field, enter the name of the aspect you want to remove from the property.
 - b. In the **Equipment Instance or Class** field, click the ellipsis button, and then select an instance or class.
 - c. In the **propertyName** field, enter the name of the property whose aspect you want to verify.
 10. Click **Save**.
 11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

RemovePropertyFromClass (IEquipment)

Allows you to remove a property from an equipment class.

On the activity configuration panel, you can use a bind or expression to define your input and output parameters.

Inputs	Data Type	Description
Equipment Class	DirectoryResource	Specifies the equipment class that the property is being removed from.
Property Name	String	Specifies the display name of the class property that is being removed.

RemovePropertyFromEquipment (IEquipment)

Allows you to remove a property from an equipment instance.

On the activity configuration panel, you can use a bind or expression to define your input and output parameters.

Inputs	Data Type	Description
Equipment Instance	DirectoryResource	Specifies the equipment instance that the property is being removed from.
Property Name	String	Specifies the display name of the instance property that is being removed.

SetDataSourceLink (IEquipment)

Sets a data source binding on a given equipment property.

Inputs	Data Type	Description
Data Source	DataAccessEndPoint	Specifies the data source link to apply.
Equipment Instance	DirectoryResource	Specifies the equipment instance. In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configura-

Inputs	Data Type	Description
		tion panel, click the ellipsis button to select a resource.
propertyName	String	Specifies the name of the property whose data source link is being changed.

No data is returned.

WriteEquipmentValue (IEquipment)

Writes the value to the Value attribute of the specified property of the selected equipment instance or class.



Note:

- You can write a value of a different data type than the actual type of the equipment property and Workflow will attempt to convert the value to the correct type. However, some data types cannot be converted. If a data type cannot be converted, an error message appears.
- If you override a default value on an property inherited from an equipment class, the new value appears in blue.

Inputs	Data Type	Description
Equipment Instance	DirectoryResource	Specifies the equipment resource to write to. In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.
Property Name	String	Specifies the name of the property attribute to write to.

Inputs	Data Type	Description
		The format for a property attribute is: propertyname.attribute.
Value	Object	Specifies the new value to write to the attribute.

No data is returned.

WritePropertyAttribute (IEquipment)

Writes to an attribute of an equipment instance or class property.

Inputs	Data Type	Description
Attribute Name	String	Specifies the attribute to write to.
Equipment Instance or Class	DirectoryResource	Specifies the equipment instance or class to write to.
Property Name	String	Specifies the name of the property whose attribute you want to write to.
Value	Object	Specifies the value to write to the attribute.



Note:

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Write an equipment property attribute

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. From the Workflows list, select the resource you want to work with.



Note:

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IEquipment** to display the list of available call methods.
 8. Click **WritePropertyAttribute**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
 9. Configure the call method:
 - a. In the **attribute** field, enter the name of the attribute you want to write to.
 - b. In the **Equipment Instance or Class** field, click the ellipsis button, and then select the instance or class whose property you want to write to.
 - c. In the **Property Name** field, enter the name of the property whose attribute you want to write to.
 - d. In the **Value** field, click the arrow, and then select Bind. The Select Variable dialog box appears.
The **Select Variable** dialog box appears.
 - e. Select the item to bind to, and then click **OK**.
 10. Click **Save**.
 11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

IEquipmentConfig

IEquipmentConfig provides methods for configuring the equipment model.

On the activity configuration panel, you can use a bind or expression to define your input and output parameters.

Subtopics in this section describe each service method for this service interface.

ProvisionEquipmentModel (IEquipmentConfig)

Uploads equipment data to the equipment model by processing the commands in the EquipmentProvisioning table.

Outputs	Description	
ReturnValue		Indicates whether the process succeeded or failed.

IEquipmentHistorian

IEquipmentHistorian allows you to read from and write to Historian tags.

On the activity configuration panel, you can use a bind or expression to define your input and output parameters.

Subtopics in this section describe each service method for this service interface.

GetHistorianTagAttributes (IEquipmentHistorian)

Retrieves the attributes of the Historian tag associated with the specified equipment property.

Inputs	Data Type	Description
equipmentAddress	DirectoryResource	<p>Specifies the equipment instance with the property associated with the Historian tag whose attributes you want to retrieve.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: In the Configure Method dialog box, click Next to </div>

Inputs	Data Type	Description
		 select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.
propertyName	String	Specifies the name of the equipment property associated with the Historian tag whose attributes you want to retrieve.

Outputs	Description
TagAttributes	Returns a collection of the Historian tag attributes.

Retrieve and configure Historian tag attributes associated with an equipment property

The following procedure describes how to retrieve and configure the attributes of an Historian tag that is associated with an equipment property.

Procedure

1. Add a workflow that will be used to configure the attributes of an Historian tag.



Note:

Activities can be inserted at both the workflow and subprocess levels.

2. In the navigator, click **Workflow > Workflows**.
3. In the Workflows list, select the resource you want to work with



Note:

A resource must already have been added in order to select it from this list.

4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.

6. Insert an activity:

◦ At the workflow level:

- From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

◦ At the subprocess level:

- Double-click the local subprocess you want to add the activity to.
- From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

7. On the activity configuration panel, click **Configure Method**.

The **Configure Method** dialog box appears, displaying a list of the available call method service providers.

8. Double-click **EquipmentHistorian** to display the list of available call methods.9. Click **GetHistorianTagAttributes**, and then click **Finish**.

10. Configure the call method as follows:

- a. In the **Equipment Address** field, click the ellipsis button, and then select the equipment instance with the property associated with the tag whose attributes you want to retrieve.
- b. In the **Property Name** field, enter the name of the equipment property associated with the tag whose attributes you want to retrieve.

11. For each tag attribute you want to change, insert a Write activity into the workflow, after the call method, and then configure it as follows:

- a. On the activity configuration panel, click the ellipsis button.

The **Select Variable** dialog box appears.

- b. Expand the call method you created in step 2 and, from the Return Value list for the call method, select the attribute you want to configure.

- c. Click **OK**.

- d. On the activity configuration panel, enter a value for the attribute or bind the value to another attribute.

12. Repeat steps 2 through 4.

13. Click **SetHistorianTagAttributes**, and then click **Finish**.

14. Configure the call method as follows:

- a. In the **Equipment Address** field, click the ellipsis button, and then select the same equipment resource you selected in step 6a.
 - b. In the **Property Name** field, enter the name of the same equipment property you specified in step 6b.
 - c. In the **Tag Attributes** field, click the arrow, and then select **Bind**.
The **Select Variable** dialog box appears.
 - d. Expand the call method you created in step 2, and then select the Return Value output parameter.
 - e. Click **OK**.
15. Click **Save**.
 16. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

IsHistorianProperty (IEquipmentHistorian)

Verifies whether the selected property is associated with an Historian tag.

Inputs	Data Type	Description
equipmentAddress	DirectoryResource	<p>Specifies the equipment instance that owns the property you want to verify.</p> <div data-bbox="1024 1304 1417 1749" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.</p> </div>

Inputs	Data Type	Description
propertyName	String	Specifies the property you want to verify as being associated with an Historian tag.

Outputs	Description
Boolean	Indicates whether the selected property is associated with an Historian tag.

ReadHistoricalValue (IEquipmentHistorian)

Retrieves historical data from the Historian tag associated with the specified equipment property.

Inputs	Data Type	Description
calculationMode	HistorianCalculationModes	Specifies the Historian calculation mode to use to filter the data to retrieve.
direction	HistorianDirection	Specifies the Historian direction to use to filter the data to retrieve.
endTime	DateTime	Specifies the end time to use to filter the data to retrieve.
equipmentAddress	DirectoryResource	Specifies the equipment instance that owns the property you want to retrieve historical data from.

 **Note:**
In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the

Inputs	Data Type	Description
		 ellipsis button to select a resource.
interval	UInt32	Specifies the number of intervals within the specified start and end times to use to filter the data to retrieve.
numberOfSamples	UInt32	Specifies the number of data samples you want to retrieve within the specified start and end times.
propertyName	String	Specifies the property you want to retrieve historical data from.
samplingMode	HistorianSamplingModes	Specifies the Historian sampling mode to use to filter the data to retrieve.
startTime	DateTime	Specifies the start time to use to filter the data to retrieve.

Outputs	Description
ProcessValue[]	Returns the historical data for the specified equipment property.

SetHistorianTagAttributes (IEquipmentHistorian)

Configures the Historian tag attributes for the specified equipment property.

Inputs	Data Type	Description
equipment Address	DirectoryResource	Specifies the equipment instance that owns the property associated with the Historian tag whose attributes you want to configure.

Inputs	Data Type	Description
		<div style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px; background-color: #e6f2ff;">  Note: In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource. </div>
propertyName	String	Specifies the name of the equipment property associated with the Historian tag whose attributes you want to configure.
tagAttributes	TagAttributes	<p>Specifies the tag attributes you want to configure.</p> <p>It is recommended that you modify only the following tag attributes for proper server to server tag setup for data collection:</p> <ul style="list-style-type: none"> • CollectorName • CollectorType • SourceAddress • CalculationTriggers • CollectionType • CollectionInterval • DataType • StoreMilliseconds
Outputs	Description	
Response	Returns the historical data for the specified equipment property.	

SetHistorianTagValue (IEquipmentHistorian)

Configures the value of the Historian tag associated with the specified equipment property.

Inputs	Data Type	Description
equipmentAddress	DirectoryResource	<p>Specifies the equipment instance that owns the property associated with the Historian tag whose value you want to configure.</p> <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.</p> </div>
propertyName	String	Specifies the name of the equipment property associated with the Historian tag whose value you want to configure.
quality	PlatformQuality	Specifies the quality label to assign to the Historian tag.
timestamp	DateTime	Specifies the date and time to assign to the Historian tag value.
value	Object	Specifies the value to write to the Historian tag.

Outputs	Description
Response	Configures the value of the Historian tag associated with the specified equipment property.

IEvent

IEventManager allows you to read from the Event model and to create an event subscription for a specific event resource.

On the activity configuration panel, you can use a bind or expression to define your input and output parameters.

GetEventSubscription (IEvent)

Creates an event subscription for either a condition or time event.

Inputs	Data Type	Description
eventAddress	DirectoryResource	Specifies the address of the event resource to create a subscription for. In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.
itemName	String	Depends on the event selected. It can be the property name, tag name, or other. In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description
ISubscription	Returns a resource that can be used to subscribe to the event.

IExtensionData

IExtensionData allows you to access temporary workflow data that was previously stored in a system repository. This stored data can be associated with both internal and external systems.

For example, if you have a workflow that contains production data that is relevant to external systems only, you can store it as extension data. This data can be retrieved and returned to the external system at a later time when production results are being output.

On the activity configuration panel, you can use a *bind* or *expression* to define your input and output parameters.

Subtopics in this section describe each service method for this service interface.

GetExtensionData (IExtensionData)

Gets extension data.

Inputs	Data Type	Description
key	String	Specifies the user-defined data extension key that was used to store the extension data.

Outputs	Description
String	Displays extension data that is retrieved.

RemoveExtensionData (IExtensionData)

Removes extension data.

Inputs	Data Type	Description
key	String	Specifies the user-defined data extension key that was used to store the extension data.

StoreExtensionData (IExtensionData)

Stores extension data.

Inputs	Data Type	Description
dataString	String	Specifies extension data.
key	String	Specifies the user-defined data extension key, which is predictable and unique. The key is used in the GetExtensionData and RemoveExtensionData call methods.

IITService

IITService allows you to read from the Proficy System model.

On the activity configuration panel, you can use a *bind* or *expression* to define your input and output parameters.

Subtopics in this section describe each service method for this service interface.

GetAllComputers (IITService)

Returns configuration data for all computers on your network.



Note:

There are no input parameters for this call method.

Outputs	Description
List<ComputerDataObject>	<p>Returns a list of computers and their configuration data.</p> <p>To access this data, use an <i>Index Activity</i> or use a <i>For Each Activity</i> to loop through all of the computer data objects.</p>

GetComputer (IITService)

Retrieves configuration data for a specified computer.

Inputs	Data Type	Description
computerName	String	Specifies the name of the computer you want to retrieve configuration data for. In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description
ComputerDataObject	Provides configuration information about the specified computer. User-defined properties and attributes are returned.



Important:

The returned collection of data is based on user-defined values and is dependent on your published data. For information on the data that can be returned, see *Automatic Login and Logout* and *Client Access Configuration* in *Startup Options*.

GetLinkedEquipment (IITService)

Reads the linked equipment address of a computer.

Inputs	Data Type	Description
computerAddress	DirectoryResource	Specifies the location of the computer linked to the selected equipment resource. In the Inputs section of the activity configuration panel, click the ellipsis button to open the Uni-

Inputs	Data Type	Description
		versal Browser to select the resource you want to use.

Outputs	Description
ReturnValue	Returns the address of the computer linked to the selected equipment resource.

ILicense

The ILicense service method provides access to active licensing information.

On the activity configuration panel, you can use a *bind* or *expression* to define your input parameters. Additionally, you can type these input parameters in manually.

Subtopics in this section describe each service method for this service interface.

GetLicenseUsage (ILicense)

Gets all the active license information.

There are no input parameters to configure for this method.

Outputs	Description
LicenseInfo[]	Consists of the Login Name, Workstation Name, and Expiry Date.

GetLicenseUsageForComputer (ILicense)

Gets all the active license information for a specified computer.

Inputs	Data Type	Description
Workstation Name	String	Specifies the computer name to obtain license information for.

Outputs	Description	
LicenseInfo[]	Returns a string array consisting of the Login Name, Workstation Name, and Expiry Date.	

GetLicenseUsageForUser (ILicense)

Gets all the active license information for a specified user.

Inputs	Data Type	Description
Login Name	String	Specifies the login name to get license information for.

Outputs	Description	
LicenseInfo[]	Returns a string array consisting of the Login Name, Workstation Name, and Expiry Date.	

IMaterial

IMaterial allows you to read from and write to the Material model.

On the activity configuration panel, you can use a *bind* or *expression* to define your input parameters. Additionally, you can type these input parameters in manually.

Subtopics in this section describe each service method for this service interface.

AddAspect (IMaterial)

Allows you to add an aspect to a material definition or class.

Inputs	Data Type	Description
aspectName	String	Specifies the name of the aspect type to add.
material	DirectoryResource	Specifies the material definition or class to add the aspect to.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the *Inputs* section of the activity configuration panel, click the ellipsis button to select a resource.

Create a material aspect

Before you begin

**Note:**

Activities can be inserted at both the workflow and subprocess levels.

About this task

Material aspects dynamically extend existing material definition instances or classes by adding new attributes and business logic without modifying or recompiling the original objects.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with.

**Note:**

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:

- At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IMaterial** to display the list of available call methods.
 8. Click **AddAspect**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
 9. Configure the call method as follows:
 - a. In the **Aspect Name** field, enter the name of the aspect you want to add to the resource.
 - b. In the **material** field, click the ellipsis button, and then select the class or definition to add the aspect to.
 10. Click **Save**.
 11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

Related reference

[AddAspect \(IMaterial\) \(on page 1777\)](#)

Related information

[Verify a material aspect \(on page 1796\)](#)

[Remove a material aspect \(on page 1816\)](#)

AddClass (IMaterial)

Allows you to add a specified material class to a specified material definition.

Inputs	Data Type	Description
Material Class	DirectoryResource	Specifies the material class to associate with the material definition.
Material Definition	DirectoryResource	Specifies the material definition to associate to the material class.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

AddLotToAssembly (IMaterial)

Adds a parent/child assembly relationship between two material lots.

Inputs	Data Type	Description
Child Lot ID	GUID	Specifies the ID of the material lot you want to add as a child to the material lot defined in the Parent Lot ID field.
Parent Lot ID	GUID	Specifies the material lot you want to define as the parent for the material lot defined in the Child Lot ID field.

Outputs	Description
	Boolean Returns <i>True</i> if the assembly relationship is created; returns <i>False</i> if the assembly relationship pre-exists.

Create a material lot assembly relationship

You can create a parent/child relationship between material lots for an assembly.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with.



Note:

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IMaterial** to display the list of available call methods.
 8. Click **CreateMaterialLot**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
 9. Configure the input parameters for either the material lot as follows:
 - a. **Optional:** In the **Description** field, enter identifying information about the material lot.
 - b. In the **Location** field, click the ellipsis button, and then select the equipment instance to associate with the material lot.
 - c. In the **Material Definition** field, click the ellipsis button, and then select the material definition to associate with the material lot.
 - d. In the **Quantity** field, enter the number of material items to create for the lot.
 - e. In the **Status** field, enter the run-time status of the material lot.
 - f. In the **UoM (Unit of Measure)** field, enter the unit of measure for the material lot.
 10. Repeat steps 8-9 to create another material lot.

11. On the activity configuration panel, click **Configure Method**.

The *Configure Method* dialog box appears, displaying a list of the available call method service providers.

12. Double-click **IMaterial** to display the list of available call methods.

13. Click **AddLotToAssembly**, and then click **Finish**.

The input parameters appear in the activity configuration panel.

14. Bind the Child Lot ID to a material lot as follows.

- a. In the Child Lot ID field, click the arrow.

The **Select Variable** dialog box appears.

- b. Select the return value of the call method you added in step 11, and then click **OK**.

15. Bind the Parent Lot ID to a material lot as follows:

- a. In the Parent Lot ID field, click the arrow.

The **Select Variable** dialog box appears.

- b. Select the return value of the call method you added in step 11, and then click **OK**.

16. Click **Save**.

17. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

AddPropertyAspect (IMaterial)

Adds an aspect to a material definition or class property.

Inputs	Data Type	Description
aspectName	String	Specifies the property aspect to associate with the property.
material	DirectoryResource	Specifies the material definition or class to add the property aspect to.
propertyName	String	Specifies the name of the property to add the aspect to.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Create a material property aspect

Before you begin

**Note:**

Activities can be inserted at both the workflow and subprocess levels.

About this task

Material property aspects dynamically extend material definition instance or class properties without modifying or recompiling the original objects.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with.

**Note:**

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.

The **Configure Method** dialog box appears, displaying a list of the available call method service providers.

7. Double-click **IMaterial** to display the list of available call methods.

8. Click **AddPropertyAspect**, and then click **Finish**.

The input parameters appear in the activity configuration panel.

9. Configure the call method as follows:

- a. In the **Aspect Name** field, enter the name of the aspect you want to add to the resource.
- b. In the **Material** field, click the ellipsis button, and then select a class or definition.
- c. In the **Property Name** field, enter the name of the property you want to add the property aspect to.

10. Click **Save**.

11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

AddPropertyToClass (IMaterial)

Allows you to add a property to a material class.

On the activity configuration panel, you can use a *bind* or *expression* to define your input parameters.

Additionally, you can type these input parameters in manually.

Inputs	Data Type	Description
Default Value	Object	Specifies the default variable value of the new material class property.
Description	String	Specifies the description of the class property.
Material Class	DirectoryResource	Specifies the material class to add the property to.
Property Name	String	Specifies a display name for the class property.
Property Type	ParameterDataType	Specifies a data type for the class property.
Unit of Measure	String	Specifies a unit of measure for the class property.

ClearPropertyFromLotProperty (IMaterial)

Clears the material definition property from being a lot property.

Inputs	Data Type	Description
materialDefinitionId	DirectoryResource	Specifies the material definition with the property you want to clear.
propertyName	String	Specifies the name of the property you want to clear.



Note:

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

CreateMaterialClass (IMaterial)

Allows you to create a new material class.

On the activity configuration panel, you can use a *bind* or *expression* to define your input parameters. Additionally, you can type these input parameters in manually.

Inputs	Data Type	Description
Description	String	Specifies the description of the new material class.
S95 ID	String	Specifies the ID of the new material class.

Outputs	Description
DirectoryResource	The new material class.

CreateMaterialDefinition (IMaterial)

Creates a new material definition.

Inputs	Data Type	Description
Description	String	Specifies the description of a new material definition.
S95 ID	String	Specifies the ID of the new material definition.

Outputs	Description
DirectoryResource	Returns the material definition that was created.

CreateMaterialLot (IMaterial)

Creates a lot for a particular material resource.

Inputs	Data Type	Description
Description	String	Specifies the description of a material resource lot.
Location	DirectoryResource	Specifies the equipment resources where the lot is located.
Material Definition	DirectoryResource	Specifies the material definition that the lot is created for.
Quantity	Double	Specifies the material quantity in a lot.
S95 ID	String	Specifies the ID of the new material lot.
Status	String	Specifies the run time status of the material lot.
UoM (Unit of Measure)	String	Specifies the unit of measure of the lot's quantity value.

Outputs	Description	
GUID		Returns a lot for a particular material resource.

CreateMaterialSublot (IMaterial)

Creates a subplot for a particular material that is under a specified parent material resource.

Inputs	Data Type	Description
Description	String	Specifies the description of a material resource subplot.
Location	DirectoryResource	Specifies the equipment resources where the subplot is located.
Lot ID	GUID	Specifies the material lot or subplot to create the subplot under.
Quantity	Double	Specifies the material quantity in a subplot.
S95 ID	String	Specifies the ID of the new material subplot.
Status	String	Specifies the run time status of the material subplot.
UoM (Unit of Measure)	String	Specifies the unit of measure of the subplot's quantity value.

Outputs	Description	
GUID		Returns a lot for a particular material resource.

DecrementMaterialLotOrSublotQuantity (IMaterial)

Reduces the quantity of a specified material lot or subplot.

Inputs	Data Type	Description
Lot or Sublot ID	GUID	Specifies the ID material lot or subplot that is being modified.
Lot Type	LotType	Specifies whether to modify a lot or a subplot.
Quantity	Double	Specifies the amount that a material lot or subplot is reduced.

Outputs	Description
Double	Returns the quantity of the lot or subplot after it has been decremented.

DeleteMaterialClass (IMaterial)

Allows you to delete a material class.

On the activity configuration panel, you can use a *bind* to define your input parameters.

Inputs	Data Type	Description
Material Class	DirectoryResource	Specifies the material class to delete.

DeleteMaterialDefinition (IMaterial)

Deletes a material definition.

Inputs	Data Type	Description
Material Definition	DirectoryResource	Specifies the material definition to delete. <div data-bbox="1024 1654 1421 1881" style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: When a resource is deleted, it cannot be recovered. </div>

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

DeleteMaterialLotOrSublot (IMaterial)

Deletes a specified material lot or subplot.

Inputs	Data Type	Description
Lot or Sublot ID	GUID	Specifies the ID material lot or subplot that is being modified.
Lot Type	LotType	Specifies whether to modify a lot or a subplot.

FindMaterialDefinitions (IMaterial)

Finds all of the material definitions that exactly match the defined parameter names and values.

The material properties you search for include those properties belonging to material definitions and classes associated with material definitions.

When you configure the search parameters, keep the following in mind:

- You can use empty strings as parameter values; however, null values are not allowed.
- You can enter a maximum of 10 parameters in each field.
- You must enter the same number of name and value parameters.
- Boolean values must be entered as 1 to represent True or Yes, or 0 to represent False or No.
- DateTime values must be in UTC time zone, with the following format: yyyy-mm-dd hh:mi:ss.
- The search returns a maximum of 5,000 results.

Property	Value	Description
Parameter names	String[]	Specifies the parameter names to use as search criteria. Click the ellipsis button to open the String Collection Editor and enter the parameter names.

Property	Value	Description
Parameter values	String[]	Specifies the parameter values to use as search criteria. Click the ellipsis button to open the String Collection Editor and enter the parameter values.

Outputs	Description
String[]	Returns an array of material definition IDs that exactly match the defined parameter names and values.

Search for material definitions using a workflow call method

This procedure describes how to search for material definitions using parameter names and values as search criteria.

About this task

When you configure the search parameters, keep the following in mind:

- You can use empty strings as parameter values; however, null values are not allowed.
- You can enter a maximum of 10 parameters in each field.
- You must enter the same number of name and value parameters.
- Boolean values must be entered as 1 to represent True or Yes, or 0 to represent False or No.
- DateTime values must be in UTC time zone, with the following format: yyyy-mm-dd hh:mi:ss.
- The search returns a maximum of 5,000 results.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with.



Note:

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.

4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IMaterial** to display the list of available call methods.
 8. Click **FindMaterialDefinitions**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
 9. In the **Parameter name** field, click the ellipsis button to open the **String Collection Editor** dialog box, and then enter the parameter names you want to use as search criteria.

**Note:**

Enter a maximum of 10, one on each line.

10. In the **Parameter values** field, click the ellipsis button to open the **String Collection Editor** dialog box, and then enter the parameter values you want to use as search criteria.

**Note:**

Enter a maximum of 10, one on each line.

11. Click **OK**, and then click **Save**.
12. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

GetAspectTypes (IMaterial)

Returns the applicable and applied aspect types for a material definition or class.

Inputs	Data Type	Description
material	DirectoryResource	Specifies the material definition or class to read.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description
ExtensionInfo[]	Returns an array of possible aspect information containing aspect name, description, and a flag indicating whether the aspect is applied to the specified resource.

Get material aspect types

Before you begin

**Note:**

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with.

**Note:**

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:

- At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IMaterial** to display the list of available call methods.
 8. Click **GetAspectTypes**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
 9. In the **Material** field, click the ellipsis button, and then select the class or definition to read.
 10. Click **Save**.
 11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

GetClassProperties (IMaterial)

Returns a list of properties defined by a material class.

Inputs	Data Type	Description
Material Class	DirectoryResource	<p>Specifies the material class that owns the properties.</p> <p>In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.</p>

Outputs	Description	
String[]		Returns an array of class property names.

GetPropertyAspectTypes (IMaterial)

Returns the applicable and applied aspect types for a property owned by a material definition or material class.

Inputs	Data Type	Description
material	DirectoryResource	Specifies the material definition or class to read.
propertyName	String	Specifies the name of the property to read.



Note:

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description	
ExtensionInfor[]		Returns an array of possible aspect information containing aspect name, description, and a flag indicating whether the aspect is applied to the specified resource.

Get material property aspect types

Before you begin



Note:

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with.

**Note:**

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IMaterial** to display the list of available call methods.
 8. Click **GetPropertyAspectTypes**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
 9. Configure the call method as follows:
 - a. In the **Material** field, click the ellipsis button, and then select a class or definition.
 - b. In the **Property Name** field, enter the name of the property to read.
 10. Click **Save**.
 11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

HasAspect (IMaterial)

Checks whether a material definition or class has the specified aspect applied to it.

Inputs	Data Type	Description
aspectName	String	Specifies the aspect type to check.
material	DirectoryResource	Specifies the material definition or class to check.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description
Boolean	True if the selected resource is augmented by the specified aspect; otherwise, False.

Verify a material aspect

Before you begin

**Note:**

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with.

**Note:**

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:

- At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IMaterial** to display the list of available call methods.
 8. Click **HasAspect**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
 9. Configure the call method as follows:
 - a. In the **Aspect Name** field, enter the name of the aspect you want to verify.
 - b. In the **Material** field, click the ellipsis button, and then select a class or definition.
 10. Click **Save**.
 11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

HasClass (IMaterial)

Checks if a given material definition is associated with a given material class.

Inputs	Data Type	Description
Material Class	DirectoryResource	Specifies the material class to look for.
Material Definition	DirectoryResource	Specifies the material definition to check.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description	
Boolean		True if the material definition is being augmented by the class; otherwise, False.

HasPropertyAspect (IMaterial)

Checks if a property owned by a material definition or material class has a specified aspect applied to it.

Inputs	Data Type	Description
aspectName	String	Specifies the aspect type to check.
material	DirectoryResource	Specifies the material definition or class to check.
propertyName	String	Specifies the name of the property to check.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description	
Boolean		True if the resource property is augmented by the specified aspect; otherwise, False.

Verify a material property aspect

Before you begin**Note:**

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with.

**Note:**

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
7. Double-click **IMaterial** to display the list of available call methods.
8. Click **HasPropertyAspect**, and click **Finish**.
The input parameters appear in the activity configuration panel.
9. Configure the call method as follows:
 - a. In the **Aspect Name** field, enter the name of the aspect you want to verify.
 - b. In the **Material** field, click the ellipsis button, and then select a class or definition.
 - c. In the **Property Name** field, enter the name of the property

10. Click **Save**.

11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

IncrementMaterialLotOrSublotQuantity (IMaterial)

Increases the quantity of a specified material lot or subplot.

Inputs	Data Type	Description
Lot or Sublot ID	GUID	Specifies the ID material lot or subplot that is being modified.
Lot Type	LotType	Specifies whether to modify a lot or a subplot.
Quantity	Double	Specifies the amount that a material lot or subplot is increased.

Outputs	Description
Double	Returns the quantity of the lot or subplot after it has been incremented.

IsLotAssemblyDescendant (IMaterial)

Returns a value indicating whether the defined material lot is the descendant of a root material lot within an assembly.

Property	Value	Description
Descendant Lot ID	GUID	Specifies the ID of the assembly material lot whose relationship with the root material lot defined in the Lot ID field you want to check.
Lot ID	GUID	Specifies the ID of the root material lot whose relationship with the assembly material lot defined

Property	Value	Description
		in the Descendant Lot ID field you want to check.
Max Depth	Int32	Specifies the maximum number of relationship levels to check between the defined assembly and root material lots.

Outputs	Value	Description
	Boolean	Returns <i>True</i> if the assembly material lot and root material lot are related, within the defined maximum depth; returns <i>False</i> if there is no relationship between the specified material lots within the defined maximum depth.

QueryMaterialLots (IMaterial)

Performs a query against a runtime database for material lots. All material lots that match the specified query criteria are returned.

In the **Configure Method** dialog box, click **Next** to select query criteria to include as input parameters on the activity configuration panel.



Note:

If you do not select any query criteria as input parameters, your query returns up to 500 of the most recently modified lots. To return additional results, configure one or more queries using the Last Modified Time (Max) and Last Modified Time (Min) parameters. Bind the Last Modified Time (Max) parameter of subsequent queries to the OldestLastModified property of the MaterialLotCollection output parameter (in the Select Variable dialog box) from the previous query to return the next set of results. Each query returns up to 500 results.

Inputs	Data Type	Description
Detail Level	MaterialLotDetailLevel	<p>Specifies the level of detail that you want the query to return.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • OnlyAttributes • IncludeChildSublots • IncludeProperties • FullDetails
Is Closed	Boolean	<p>Indicates whether the returned material lots are closed or finished. Valid values are:</p> <ul style="list-style-type: none"> • True • False
Last Modified Time (Max)	DateTime	<p>Specifies the latest modified date and time to include in your query. This is the most recent date/time that returned material lots were modified.</p>
Lsat Modified Time (Min)	DateTime	<p>Specifies the earliest modified date and time to include in your query. This is the earliest date/time that returned material lots were modified.</p>
Location	DirectoryResource	<p>Specifies the equipment location to include in your query.</p>
Material Definition	DirectoryResource	<p>Specifies the material definition to include in your query.</p>
Quantity Max	Double	<p>Specifies the maximum quantity to return.</p>
Quantity Min	Double	<p>Specifies the minimum quantity to return.</p>

Inputs	Data Type	Description
S95 ID	String	Specifies the name of the material lot to include in your query. <div data-bbox="1024 390 1419 743" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;">  Note: The name you enter must be an exact match for the S95 ID. S95 IDs are case sensitive and wild cards are not permitted in a query. </div>
Status	String	Specifies the run time status to include in your query.

Outputs	Description
MaterialLotCollection	Returns a collection of material lots for a specific material resource based on the criteria indicated.

QueryMaterialSublots (IMaterial)

Performs a query against a runtime database for material sublots. All material sublots that match the specified query criteria are returned.

In the **Configure Method** dialog box, click **Next** to select query criteria to include as input parameters on the activity configuration panel.



Note:

If you do not select any query criteria as input parameters, your query returns up to 500 of the most recently modified sublots. To return additional results, configure one or more queries using the Last Modified Time (Max) and Last Modified Time (Min) parameters. Bind the Last Modified Time (Max) parameter of subsequent queries to the OldestLastModified property of the MaterialLotCollection output parameter (in the Select Variable dialog box) from the previous query to return the next set of results. Each query returns up to 500 results.

Inputs	Data Type	Description
Detail Level	MaterialLotDetailLevel	<p>Specifies the level of detail that you want the query to return.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • OnlyAttributes • IncludeChildSublots • IncludeProperties • FullDetails
IsClosed	Boolean	<p>Indicates whether the returned material sublots are closed or finished. Valid values are:</p> <ul style="list-style-type: none"> • True • False
Last Modified Time (Max)	DateTime	<p>Specifies the latest modified date and time to include in your query. This is the most recent date/time that returned material sublots were modified.</p>
Last Modified Time (Min)	DateTime	<p>Specifies the earliest modified date and time to include in your query. This is the earliest date/time that returned material sublots were modified.</p>
Location	DirectoryResource	<p>Specifies the equipment location to include in your query.</p>
Material Definition	DirectoryResource	<p>Specifies the material definition to include in your query.</p>
Material Lot	GUID	<p>Specifies the ID of the material lot to include in your query.</p>
Quantity Max	Double	<p>Specifies the maximum quantity to return.</p>

Inputs	Data Type	Description
Quantity Min	Double	Specifies the minimum quantity to return.
S95 ID	String	Specifies the name of the material subplot you are querying. <div data-bbox="1024 495 1419 852" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;">  Note: The name you enter must be an exact match for the S95 ID. S95 IDs are case sensitive and wild cards are not permitted in a query. </div>
Status	String	Specifies the run time status to include in your query.

Outputs	Description
MaterialSublotCollection	Returns a collection of material sublots for a specific material lot based on the criteria indicated. To access these properties, use an <i>Index Activity</i> or a <i>For Each</i> loop. You can use the parameter name as the index for the Index activity.

ReadAllMaterialValues (IMaterial)

Returns the values of all of the properties and attributes of a material resource.

Inputs	Data Type	Description
Material Address	DirectoryResource	Specifies the material resource to read.

Outputs	Description	
NameDataValueCollection		Returns a list of the user-defined values of the properties and attributes for the specified resource. To access these properties, use an <i>Index Activity</i> or a <i>For Each</i> loop. You can use the parameter name as the index for the Index activity.

**Important:**

The returned collection of data is based on user-defined values and is dependent on your published data. For information on the attributes that can be read from, see the related properties for the material model.

ReadAllPropertyAttributes (IMaterial)

Returns the values of all attributes of a material definition or class property.

Inputs	Data Type	Description
material	DirectoryResource	Specifies the material definition or material class to read.
propertyName	String	Specifies the name of the property whose attributes you want to read.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description	
AttributeCollection		Returns a list of the user-defined values of the attributes for the specified property.

Read multiple material property attributes

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with.



Note:

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
7. Double-click **IMaterial** to display the list of available call methods.
8. Click **ReadAllPropertyAttributes**, and then click Finish. The input parameters appear in the activity configuration panel.
9. Configure the call method as follows:
 - a. In the **Material** field, click the ellipsis button, and then select a resource.
 - b. In the **Property Name** field, enter the name of the property whose attributes you want to read.
10. Click **Save**.
11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

ReadClasses (IMaterial)

Returns a list of classes associated with a specified material definition.

Inputs	Data Type	Description
Material Definition	DirectoryResource	Specifies the material definition to read classes from. In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description
DirectoryResource[]	Returns an array of material classes associated with the specified material definition.

ReadClassProperties (IMaterial)

Retrieves the property values belonging to a specified class from a specified material definition.

Inputs	Data Type	Description
Material Class	DirectoryResource	Specifies the material class that defines the properties to read.
Material Definition	DirectoryResource	Specifies the material definition to read the material class properties from.



Note:

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description	
NameDataValueCollection		Returns a list of the user-defined values from the specified material definition for the class properties.

ReadMaterialLot (IMaterial)

Returns a list of attributes associated with a specified material lot.

Inputs	Data Type	Description
Detail Level	MaterialLotDetailLevel	Specifies the level of detail that you want the query to return. Valid values are: <ul style="list-style-type: none"> • OnlyAttributes • IncludeChildSublots • IncludeProperties • FullDetails
Lot ID	GUID	Specifies the material lot to read.

Outputs	Description	
MaterialLotAttributes		Returns a collection of material lot attributes.

ReadMaterialSublot (IMaterial)

Returns a list of attributes associated with a specified material sublot.

Inputs	Data Type	Description
Detail Level	MaterialLotDetailLevel	Specifies the level of detail that you want the query to return. Valid values are:

Inputs	Data Type	Description
		<ul style="list-style-type: none"> • OnlyAttributes • IncludeChildSublots • IncludeProperties • FullDetails
Sublot ID	GUID	Specifies the material sublot to read.

Outputs	Description
MaterialSublotAttributes	Returns a collection of material lot attributes.

ReadMaterialValue (IMaterial)

Reads the value of a single property or attribute of a specific material resource.



Note:

In the Edit Method dialog box, click Next to add class definitions to the call method. This returns only the parameters you choose from the class.

Inputs	Data Type	Description
Material Address	DirectoryResource	<p>Specifies the material resource to read.</p> <p>In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.</p>
propertyName	String	<p>Specifies the property and/or attribute names to return.</p> <p>The format for a property is: <code>propertyName</code>.</p>

Inputs	Data Type	Description
		<p>The format for a property and attribute is: propertyname.attribute.</p> <p>Use a comma to separate each propertyName you want to retrieve; for example, propertyname1, propertyname2.attribute1, propertyname3.attribute1.</p>

Outputs	Description
DataValue	Returns all data about the specified property and/or attribute for the selected resource.

ReadMaterialValuesByName (IMaterial)

Returns the values of all of the named properties and attributes of a material resource.



Note:

In the Edit Method dialog box, click Next to add class definitions to the call method. This returns only the parameters you choose from the class.

Inputs	Data Type	Description
Material Address	DirectoryResource	<p>Specifies the material resource to read.</p> <p>In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.</p>
propertyNames	String[]	Specifies the names of the attributes or properties to read.

Inputs	Data Type	Description
		In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description
NameDataValueCollection	<p>Returns a list of the user-defined values of the named properties and attributes for the specified resource.</p> <p>To access these properties, use an <i>Index Activity</i> or a <i>For Each</i> loop. You can use the parameter name as the index for the Index activity.</p>



Important:

The returned collection of data is based on user-defined values and is dependent on your published data. For information on the attributes that can be read from, see the related properties for the material model.

ReadPropertyAttribute (IMaterial)

Returns the value of an attribute of a material definition or class property.

Inputs	Data Type	Description
attribute	String	Specifies the property attribute to read.
material	DirectoryResource	Specifies the material definition or class to read.

Inputs	Data Type	Description
propertyName	String	Specifies the name of the property whose attribute you want to read.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description
Object	Returns the user-defined value of the attribute for the specified property.

Read a material property attribute

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with.

**Note:**

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:

- Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IMaterial** to display the list of available call methods.
 8. Click **ReadPropertyAttribute**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
 9. Configure the call method as follows:
 - a. In the **Attribute** field, enter the name of the attribute you want to read.
 - b. In the **Material** field, click the ellipsis button, and then select the material definition or class whose property you want to read.
 - c. In the **Property Name** field, enter the name of the property whose attribute you want to read.
 10. Click **Save**.
 11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

ReadPropertyAttributesByName (IMaterial)

Returns the attribute names of the specified material definition or class property.

Inputs	Data Type	Description
attributes	String[]	Specifies the names of the attributes to read.
material	DirectoryResource	Specifies the material definition or class to read.
propertyName	String	Specifies the name of the property whose attributes you want to read.



Note:

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description	
AttributeCollection		Returns a list of the user-defined values of the attributes for the specified property.

Read multiple specific material property attributes

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with.



Note:

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
7. Double-click **IMaterial** to display the list of available call methods.
8. Click **ReadPropertyAttributesByName**, and then click Finish. The input parameters appear in the activity configuration panel.
9. Configure the call method as follows:

- a. In the **Attributes** field, click the ellipsis button to open the String Collection Editor dialog box where you can enter the names of the attributes you want to read.
 - b. In the **Material** field, click the ellipsis button, and then select the material definition or class whose property you want to read.
 - c. In the **Property Name** field, select the property whose attributes you want to read.
10. Click **Save**.
 11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

RemoveAspect (IMaterial)

Removes an aspect from a specified material definition or class.

Inputs	Data Type	Description
aspectName	String	Specifies the name of the aspect type to remove.
material	DirectoryResource	Specifies the material definition or class to remove the aspect from.



Note:

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Remove a material aspect

Before you begin



Note:

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with.

**Note:**

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IMaterial** to display the list of available call methods.
 8. Click **RemoveAspect**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
 9. Configure the call method as follows:
 - a. In the **Aspect Name** field, enter the name of the aspect you want to remove.
 - b. In the **Material** field, click the ellipsis button, and then select the class or definition to remove the aspect from.
 10. Click **Save**.
 11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

RemoveClass (IMaterial)

Removes a specified material class from a specified material definition.

Inputs	Data Type	Description
materialClass	DirectoryResource	Specifies the name of the material class to remove from the material definition.
materialDefinition	DirectoryResource	Specifies the material definition to remove the class from.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

RemoveLotFromAssembly (IMaterial)

Removes an assembly relationship between a parent material lot and a child material lot.

Inputs	Data Type	Description
Child Lot ID	GUID	Specifies the ID of the material lot whose relationship you want to remove from the material lot defined in the Parent Lot ID field.
Parent Lot ID	GUID	Specifies the ID of the material lot you whose relationship you want to remove from the material lot defined in the Child Lot ID field.

Outputs	Value	Description
	Boolean	Returns <i>True</i> if the assembly relationship is removed; returns <i>False</i> if the assembly relationship did not exist.

RemovePropertyAspect (IMaterial)

Removes an aspect from a property owned by a material definition or material class.

Inputs	Data Type	Description
aspectName	String	Specifies the name of the aspect type to remove.
material	DirectoryResource	Specifies the material definition or class to remove the aspect from.
propertyName	String	Specifies the name of the property to remove the aspect from.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Remove a material property aspect

Before you begin

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with.

**Note:**

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:

- Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IMaterial** to display the list of available call methods.
 8. Click **RemovePropertyAspect**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
 9. Configure the call method as follows:
 - a. In the **Aspect Name** field, enter a name of the aspect you want to remove.
 - b. In the **Material** field, click the ellipsis button, and then select class or definition..
 - c. In the **Property Name** field, select the property you want to remove the aspect from.
 10. Click **Save**.
 11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

RemovePropertyFromClass (IMaterial)

Allows you to remove a property from a material class.

On the activity configuration panel, you can use a bind or expression to define your input and output parameters.

Inputs	Data Type	Description
Material Class	DirectoryResource	Specifies the material class that the property is being removed from.
Property Name	String	Specifies the display name of the class property that is being removed.

SetMaterialLotOrSublotLocation (IMaterial)

Sets the equipment location for a specified material lot or subplot.

Inputs	Data Type	Description
Location	DirectoryResource	Specifies the equipment resources where the lot or subplot is located.
Lot or Sublot ID	GUID	Specifies the ID material lot or subplot that is being modified.
Lot Type	LotType	Specifies whether to modify a lot or a subplot.

SetMaterialLotOrSublotQuantity (IMaterial)

Sets the quantity specified by a lot or subplot requirement.

Inputs	Data Type	Description
Lot or Sublot ID	GUID	Specifies the ID material lot or subplot that is being modified.
Lot Type	LotType	Specifies whether to modify a lot or a subplot.
Quantity	Double	Specifies the material quantity in a lot or subplot.

SetMaterialLotOrSublotStatus (IMaterial)

Sets the status of a specified lot or subplot.

Inputs	Data Type	Description
Lot or Sublot ID	GUID	Specifies the ID material lot or subplot that is being modified.
Lot Type	LotType	Specifies whether to modify a lot or a subplot.
Status	Double	Specifies the run time status of the material lot or subplot.

SetMaterialLotToClosed (IMaterial)

Sets the specified material lot to a closed status. When a lot is in this state, it is no longer part of runtime, so it can be archived and purged.

Inputs	Data Type	Description
Lot ID	GUID	Specifies the material lot to close.

SetPropertyToLotProperty (IMaterial)

Specifies the material to be a lot property.

Inputs	Data Type	Description
materialDefinitionId	DirectoryResource	Specifies the material definition whose property you want to set as a lot property.
propertyName	String	Specifies the name of the property you want to set as a lot property.



Note:

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

UpdateMaterialLotS95Id (IMaterial)

Updates the S95 ID of a material lot.

Inputs	Data Type	Description
Lot Id	GUID	Specifies the ID of the material lot whose S95 ID you want to update.
S95Id	String	Specifies the S95 ID you want to apply to the material lot.

WriteMaterialLotProperties (IMaterial)

Writes to properties of a specified material lot.

Inputs	Data Type	Description
Lot ID	GUID	Specifies the material lot to modify.

WriteMaterialValue (IMaterial)

Writes a value to an attribute or property for a material definition.

Inputs	Data Type	Description
propertyName	String	Specifies the name of the property or attribute to write to. In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.
Resource	DirectoryResource	Specifies the material resource to update. In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.
Value	Object	Specifies the new value of the attribute or property. In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs

Inputs	Data Type	Description
		section of the activity configuration panel, click the ellipsis button to select a resource.

WritePropertyAttribute (IMaterial)

Writes to an attribute of a material definition or class property.

Inputs	Data Type	Description
attribute	String	Specifies the attribute to write to.
material	DirectoryResource	Specifies the material definition or class to write to.
propertyName	String	Specifies the name of the property whose attribute you want to write to.
Value	Object	Specifies the value to write to the attribute.



Note:

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Write a material property attribute

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with.



Note:

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:

- At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IMaterial** to display the list of available call methods.
 8. Click **WritePropertyAttribute**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
 9. Configure the call method as follows:
 - a. In the **Attribute** field, enter the name of the attribute you want to write to.
 - b. In the **Material** field, click the ellipsis button, and then select the material definition or class whose property you want to write to.
 - c. In the **Property Name** field, enter the name of the property whose attribute you want to write to.
 - d. In the **Value** field, click the arrow, and then select Bind. The Select Variable dialog box appears.
The **Select Variable** dialog box appears.
 - e. Select the item to bind to, and then click **OK**.
 10. Click **Save**.
 11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

IOpcClient

IOpcClient allows you to read from the OPC Client model.

On the activity configuration pane, you can use a *Workflow Bind* or *Workflow Expression* to define your input and output parameters.

Subtopics in this section describe each service method for this service interface.

AddOPCGroup (IOpcClient)

Adds an OPC group to the specified OPC server.

Inputs	Data Type	Description
OPC Server	DirectoryResource	Specifies the OPC server under which the group is created.
Group Name	String	Specifies the name of the OPC group.
Group Description	String	Specifies the description of the OPC group.
Update Rate (ms)	int32	Specifies the rate in milliseconds that the server is requested to update items in the group.
Enabled	Boolean	Specifies whether the group is enabled. True if so; otherwise, False. <div data-bbox="1024 1241 1419 1377" style="border: 1px solid #0070C0; border-radius: 10px; padding: 5px;">  Note: The default is True. </div>
Device Read	Boolean	Specifies whether to read directly from the device or read from the OPC server's cache. True if so; otherwise, False. <div data-bbox="1024 1598 1419 1866" style="border: 1px solid #0070C0; border-radius: 10px; padding: 5px;">  Note: The default is True. If false, reading is performed against the OPC server's cache. </div>

Inputs	Data Type	Description
Deadband (%)	Single	Specifies the minimum percentage of the value range of an item that a value in the group must change to trigger a data changed event.

Outputs	Description
Response	Returns a response indicating creation of the OPC group.

AddOPCItem (IOpcClient)

Adds the specified OPC item to the specified OPC group.

Inputs	Data Type	Description
Group Address	DirectoryResource	Specifies the address of the OPC group to which the item is added.
Item Name	String	Specifies the name of the OPC item.

Outputs	Description
OpcltemStatus	Returns the status of the added item.

CreateDataSourceLink (IOpcClient)

Creates a data source link that can be used to access an OPC item.

Inputs	Data Type	Description
Group Address	DirectoryResource	Specifies the address of the OPC group containing the item.
Item Name	String	Specifies the name of the OPC item.

Outputs	Description
DataAccessEndPoint	Returns a data source link to the specified data item.

CreateDataSourceLinkWithAttributes (IOpcClient)

Create a data source link that can be used to access an OPC item.

Inputs	Data Type	Description
OPC Item	DirectoryResource	Specifies the item for which a link is created.
Value	Boolean	Includes the Item Value in the resulting link.
Quality	Boolean	Includes the Item Quality in the resulting link.
Timestamp	Boolean	Includes the Item Timestamp in the resulting link.

Outputs	Description
DataAccessEndPoint	Returns a data source link to the specified data item.

DeleteOPCItem (IOpcClient)

Deletes the specified OPC item from the specified OPC group.

Inputs	Data Type	Description
Group Address	DirectoryResource	Specifies the address of the OPC group from where the item is removed.
Item Name	String	Specifies the name of the OPC item.

Outputs	Description
Response	Returns a response indicating the removal of the item from the OPC group.

WriteGroupData (IOpcClient)

Writes to a group of OPC items simultaneously.

Inputs	Data Type	Description
Write Requests	WriteRequest[]	Specifies a collection of OPC group write requests.

IOpcUa

IOpcUa allows you to read from the OPC UA model.

On the activity configuration panel, you can use a *Workflow Bind* or *Workflow Expression* to define your input and output parameters.

Subtopics in this section describe each service method for this service interface.

AddOPCUAGroup (IOpcUa)

Adds an OPC UA group to the specified OPC UA server.

Inputs	Data Type	Description
OPC UA Server	DirectoryResource	Specifies the OPC UA server under which the group is created.
Group Name	String	Specifies the name of the OPC UA group.
Group Description	String	Specifies the description of the OPC UA group.

Outputs	Description
String	Returns a response indicating creation of the OPC UA group.

AddOPCUAItem (IOpcUa)

Adds the specified OPC UA item to the specified OPC UA group.

Inputs	Data Type	Description
OPC UA Group	DirectoryResource	Specifies the address of the OPC UA group to which the item is added.
Item Name	String	Specifies the name of the OPC UA item.

Outputs	Description
OpcUaItemStatus	Returns the status of the added item.

CreateDataSourceLink (IOpcUa)

Create a data source link that can be used to access an OPC UA item.

Inputs	Data Type	Description
Group Address	DirectoryResource	Specifies the address of the OPC UA group containing the item.
Item Name	String	Specifies the name of the OPC UA item.



Note:

Beginning with 2.6 SP1 SIM 4 release, this method accepts only two input parameters (Group Address, Item Name), which by default includes the attributes namely Value, Quality, and Timestamp in the resulting link. Prior implementations of this method will continue to function as in the past.

Outputs	Description
DataAccessEndPoint	Returns a data source link to the specified data item.

CreateDataSourceLinkWithAttributes (IOpcUa)

Create a data source link that can be used to access an OPC UA item.

Inputs	Data Type	Description
OPC UA Item	DirectoryResource	Specifies the item for which a link is created.
Value	Boolean	Includes the Item Value in the resulting link.
Quality	Boolean	Includes the Item Quality in the resulting link.
Timestamp	Boolean	Includes the Item Timestamp in the resulting link.

Outputs	Description
DataAccessEndPoint	Returns a data source link to the specified data item.

DeleteOPCUAItem (IOpcUa)

Deletes the specified OPC UA item from the specified OPC UA group.

Inputs	Data Type	Description
OPC UA Group	DirectoryResource	Specifies the address of the OPC UA group from where the item is removed.
Item Name	String	Specifies the name of the OPC UA item.

WriteGroupData (IOpcUa)

Writes to a group of OPC UA items simultaneously.

Inputs	Data Type	Description
Write Requests	WriteRequest[]	Specifies a collection of OPC UA group write requests.

IPersonnel

IPersonnel allows you to read from and write to the Personnel model.

On the activity configuration panel, you can use a *bind* or *expression* to define your input and output parameters.

Subtopics in this section describe each service method for this service interface.

AddAspect (IPersonnel)

Allows you to add an aspect to a person or personnel class.

Inputs	Data Type	Description
aspectName	String	Specifies the name of the aspect type to add.
Person or Class	DirectoryResource	Specifies the person or personnel class to add the aspect to.



Note:

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Create a personnel aspect

Before you begin



Note:

Activities can be inserted at both the workflow and subprocess levels.

About this task

Personnel aspects dynamically extend existing personnel instances or classes by adding new attributes and business logic without modifying or recompiling the original objects.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with.



Note:

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IPersonnel** to display the list of available call methods.
 8. Click **AddAspect**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
 9. Configure the call method as follows:
 - a. In the **Aspect Name** field, enter the name of the aspect you want to add to the resource.
 - b. In the **Person or Class** field, click the ellipsis button, and then select a person or class.
 10. Click **Save**.
 11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

Related information

[Remove a personnel aspect \(on page 1847\)](#)

[Verify a personnel aspect \(on page 1841\)](#)

AddClass (IPersonnel)

Associates a personnel class with a specific personnel resource.

Inputs	Data Type	Description
person	DirectoryResource	Specifies the personnel resource to read. In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.
personnelClass	DirectoryResource	Specifies the personnel class to associate with the personnel resource. In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.

AddPropertyToClass (IPersonnel)

Allows you to add a property to a personnel class.

On the activity configuration panel, you can use a *bind* or *expression* to define your input and output parameters.

Inputs	Data Type	Description
Default Value	Object	Specifies the default variable value of the new personnel class property.
Description	String	Specifies the description of the class property.
Personnel Class	DirectoryResource	Specifies the personnel class to add the property to.

Inputs	Data Type	Description
Name	String	Specifies a display name for the class property.
Type	ParameterDataType	Specifies a data type for the class property.
Unit of Measure	String	Specifies a unit of measure for the class property.

ChangeAuthenticatedPersonPassword (IPersonnel)

Changes a person's password using encrypted passwords.

Inputs	Data Type	Description
loginName	String	Specifies the login name of the person whose password is being changed.
newPassword	String	Specifies the new password.
oldPassword	String	Specifies the old password.

ChangePersonPassword (IPersonnel)

Changes a person's password using plaintext passwords.

Inputs	Data Type	Description
loginName	String	Specifies the login name of the person whose password is being changed.
newPassword	String	Specifies the new password.
oldPassword	String	Specifies the old password.

CreatePerson (IPersonnel)

Creates a new person.

Inputs	Data Type	Description
Description	String	Specifies the description of the person.
name	String	<p>Specifies the name of the person, used as additional identification for informational use.</p> <div data-bbox="1026 541 1419 1575" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Tip: If you want to create a Windows domain user, you must enter the name value in the following format: <Fully Qualified Windows Domain Name>\<Windows Account Name> (for example MyDomain.MyCompany.com\RobertGray). Because the Windows account name is not verified against the domain when creating the user, be sure to enter the correct account name; otherwise, the user will not be able to log in to Workflow. For more information, see <i>Windows Domain-Based Security</i>.</p> </div>
S95 ID	String	Specifies the S95 ID of the person to create.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description	
DirectoryResource		Returns the information related to the new person.

CreatePersonnelClass (IPersonnel)

Allows you to create a new personnel class.

On the activity configuration panel, you can use a *bind* or *expression* to define your input and output parameters.

Inputs	Data Type	Description
Description	String	Specifies the description of the new personnel class.
S95 ID	String	Specifies the ID of the new personnel class.

Outputs	Description	
DirectoryResource		The new personnel class.

DeletePerson (IPersonnel)

Deletes a person resource.

Inputs	Data Type	Description
person	DirectoryResource	Specifies the person to delete. <div data-bbox="1024 1482 1419 1709" style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; background-color: #e6f2ff;">  Note: When a resource is deleted, it cannot be recovered. </div>

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

DeletePersonnelClass (IPersonnel)

Allows you to delete a personnel class.

On the activity configuration panel, you can use a *bind* to define your input parameters.

Inputs	Data Type	Description
Personnel Class	DirectoryResource	Specifies the personnel class to delete.

GetAspectTypes (IPersonnel)

Returns the applicable and applied aspect types for a person or personnel class.

Inputs	Data Type	Description
Person or Class	DirectoryResource	Specifies the person or personnel class to read.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description
ExtensionInfo[]	Returns an array of possible aspect information containing aspect name, description, and a flag indicating whether the aspect is applied to the specified resource.

Get personnel aspect types

Before you begin



Note:

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with



Note:

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
7. Double-click **IPersonnel** to display the list of available call methods.
8. Click **GetAspectTypes**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
9. In the **Person or Class** field, select the person or class to read.

10. Click **Save**.
11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

GetClassProperties (IPersonnel)

Gets the properties for a specific class.

Inputs	Data Type	Description
personnelClass	DirectoryResource	Specifies the personnel class to read. In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description
String[]	Returns the properties of the class.

HasAspect (IPersonnel)

Checks whether a person or personnel class has the specified aspect applied to it.

Inputs	Data Type	Description
aspectName	String	Specifies the aspect type to check.
Person or Class	DirectoryResource	Specifies the person or personnel class to check.



Note:

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description	
Boolean		True if the selected resource is augmented by the specified aspect; otherwise, False.

Verify a personnel aspect

Before you begin



Note:

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with



Note:

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
7. Double-click **IPersonnel** to display the list of available call methods.

8. Click **HasAspect**, and then click **Finish**.

The input parameters appear in the activity configuration panel.

9. Configure the method as follows:

a. In the **Aspect Name** field, enter the name of the aspect you want to verify.

b. In the **Person or Class** field, click the ellipsis button, and then select a person or class.

10. Click **Save**.

11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

HasClass (IPersonnel)

Checks if a specific person is associated with a specific class.

Inputs	Data Type	Description
person	DirectoryResource	Specifies the person to check. In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.
personnelClass	DirectoryResource	Specifies the personnel class to check. In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description
Boolean	True if a specific person is being augmented by the class; otherwise, False.

ReadAllPersonnelValues (IPersonnel)

Returns the values of all of the properties and attributes of a personnel resource.

Inputs	Data Type	Description
personnelAddress	DirectoryResource	<p>Specifies the personnel resource to read.</p> <p>In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.</p>

Outputs	Description
NameDataValueClass	<p>Returns a list of the user-defined values of the properties and attributes for the specified resource.</p> <p>To access these properties, use an <i>Index Activity</i> or a <i>For Each</i> loop. You can use the parameter name as the index for the Index activity.</p>



Important:

The returned collection of data is based on user-defined values and is dependent on your published data. For information on the property attributes that can be read from, see the related properties for *Workflow Users and Personnel Classes*, *Workflow Groups*, and *Key Set Permissions*.

ReadClasses (IPersonnel)

Get the classes associated with a specific person.

Inputs	Data Type	Description
person	DirectoryResource	Specifies the personnel resource to read the classes from. In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description
DirectoryResource[]	Returns the class associated with a specific person.

ReadClassProperties (IPersonnel)

Retrieves the values for properties of a specific class from a specified person.

Inputs	Data Type	Description
person	DirectoryResource	Specifies the personnel resource to read. In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.
personnelClass	DirectoryResource	Specifies the personnel class to read the properties from. In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configura-

Inputs	Data Type	Description
		tion panel, click the ellipsis button to select a resource.

Outputs	Description
NameDataValueCollection	<p>Returns a list of the user-defined values of the properties and attributes for the specified class.</p> <p>To access these properties, use an <i>Index Activity</i> or a <i>For Each</i> loop. You can use the parameter name as the index for the Index activity.</p>



Important:

The returned collection of data is based on user-defined values and is dependent on your published data. For information on the property attributes that can be read from, see the related properties for *Workflow Users and Personnel Classes*, *Workflow Groups*, and *Key Set Permissions*.

ReadPersonnelValuesByName (IPersonnel)

Returns the values of all of the named properties and attributes of a personnel resource.

Inputs	Data Type	Description
personnelAddress	DirectoryResource	<p>Specifies the personnel resource to read.</p> <p>In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.</p>
propertyNames	String[]	Specifies the names of the properties and attributes to read.

Inputs	Data Type	Description
		In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description
NameDataValueCollection	<p>Returns a list of the user-defined values of the named properties and attributes for the specified resource.</p> <p>To access these properties, use an <i>Index Activity</i> or a <i>For Each</i> loop. You can use the parameter name as the index for the Index activity.</p>



Important:

The returned collection of data is based on user-defined values and is dependent on your published data. For information on the property attributes that can be read from, see the related properties for *Workflow Users and Personnel Classes*, *Workflow Groups*, and *Key Set Permissions*.

RemoveAspect (IPersonnel)

Removes an aspect from a person or personnel class.

Inputs	Data Type	Description
aspectName	String	Specifies the aspect type to remove.
Person or Class	DirectoryResource	Specifies the person or personnel class to remove the aspect from.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Remove a personnel aspect

Before you begin

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with

**Note:**

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
7. Double-click **IPersonnel** to display the list of available call methods.
8. Click **RemoveAspect**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
9. Configure the call method as follows:

- a. In the **Aspect Name** field, enter the name of the aspect you want to remove.
 - b. In the **Person or Class** field, click the ellipsis button, and then select the person or class to remove the aspect from.
10. Click **Save**.
 11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

RemoveClass (IPersonnel)

Removes a specified personnel class association from the selected personnel resource.

Attribute	Value	Description
person	DirectoryResource	Specifies the person to remove the class from.
personnelClass	DirectoryResource	Specifies the personnel class to remove.

RemovePropertyFromClass (IPersonnel)

Allows you to remove a property from a personnel class.

On the activity configuration panel, you can use a *bind* or *expression* to define your input and output parameters.

Inputs	Data Type	Description
Personnel Class	DirectoryResource	Specifies the personnel class that the property is being removed from.
Property Name	String	Specifies the display name of the class property that is being removed.

VerifyUserCredentials (IPersonnel)

Performs a check to verify that the provided user credentials are valid.

Inputs	Data Type	Description
Domain	String	Specifies the Windows domain of the user. Use an empty string if the provided credentials are not Windows credentials.
IsEncrypted	Boolean	Specifies whether the entered password is encrypted (<i>True</i>) or plain text (<i>False</i>).
Login Name	String	Specifies the login name of the account being verified.
Password	String	Specifies the password of the account being verified.

Outputs	Description
Boolean	<p>Indicates if the user credentials are valid.</p> <p>Valid return values are:</p> <ul style="list-style-type: none"> • true • false

WritePersonnelValue (IPersonnel)

Writes a value to an attribute or property for a personnel instance.

Inputs	Data Type	Description
personnelAddress	DirectoryResource	<p>Specifies the personnel resource to update.</p> <p>In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configura-</p>

Inputs	Data Type	Description
		tion panel, click the ellipsis button to select a resource.
propertyName	String	<p>Specifies the name of the property or attribute to write to.</p> <p>In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.</p>
Value	Object	<p>Specifies the new value of the attribute or property.</p> <p>In the Configure Method dialog box, click Next to select a resource. Otherwise, in the Inputs section of the activity configuration panel, click the ellipsis button to select a resource.</p>

IProductionModel

IProductionModel allows you to read from or write to the Production model.

On the activity configuration panel, you can use a *bind* or *expression* to define your input and output parameters.

Subtopics in this section describe each service method for this service interface.

AddChildWorkDefinitionSegment (IProductionModel)

Associates a work definition segment as a child of another work definition segment.

You can create a segment hierarchy with a maximum of four levels. A work definition segment can be associated with multiple child segments; however, the association name for each relationship must be unique.

**Important:**

A work definition segment cannot be a child of another work definition segment if it is already a parent to that segment. As well, a work definition segment cannot be a parent to another work definition segment if it is already a child of that segment.

For example, you have two parent segments, named Build Bicycle and Build Tricycle. For Build Bicycle, you have two Build Tire child segments. Even though the child segment being used is the same for both tires, you must provide a unique name to each one of the associations: Front Tire and Rear Tire. The Build Tricycle segment could also have a child segment named Front Tire. Even though the name of the association is the same as Build Bicycle, the association is unique because it is for a different parent segment.

A parent work definition segment can be a container for its child segments, or it can have its own execution steps. You can set the execution order on all levels of work definition segments within a hierarchy.

Inputs	Data Type	Description
Child	DirectoryResource	Specifies the child work definition segment you are adding to the selected parent segment.
Name	String	Specifies the name that uniquely identifies the association between the parent and child work definition segments.
Order	Int32	Specifies the order of the associated child work definition segment.



Note:
We recommend that you define the child segments' order by the execution order; that is, the order in which each child

Inputs	Data Type	Description
		 segment will be executed.
Parent	DirectoryResource	Specifies the parent work definition segment you are adding a child segment to.

AddEquipmentToProcessSegment (IProductionModel)

Adds an equipment specification to a work process segment.

The following table describes the input parameters for this method.

Inputs	Data Type	Description
Description	String	Specifies a description for the equipment specification.
Equipment	DirectoryResource	Specifies the equipment resource to add as a specification to the selected work process segment.
Name	String	Specifies a name for the equipment specification.
Quantity	Double	Specifies the quantity of the equipment specification.
Unit of Measure	String	Specifies the unit of measure of the equipment specification.
Work Process Segment	DirectoryResource	Specifies the work process segment to add the equipment specification to.

AddMasterSegmentToWorkDefinition (IProductionModel)

Adds a master segment to a work definition.

A master segment is a work process segment that contains information that applies to all work definition segments associated with a work definition. In other words, a master segment is a template for a

work request. For example, a customer name can be represented as a segment parameter in a master segment.

A master segment can be associated with multiple work definitions; however, a work definition can be associated with only one master segment. Work requests are created from an individual work definition and, therefore, can also be associated with only one master segment.

Inputs	Data Type	Description
Master Segment	DirectoryResource	Specifies the work process segment to add to the selected work definition as the master segment.
Work Definition	DirectoryResource	Specifies the work definition to add the master segment to.

AddMaterialToProcessSegment (IProductionModel)

Adds a material specification to a work process segment.

Inputs	Data Type	Description
Description	String	Specifies a description for the material specification.
material	DirectoryResource	Specifies the material resource to add as a specification to the selected work process segment.
Material Use	DirectoryResource	Specifies the material resource to add as a specification to the selected work process segment.
Name	String	Specifies a name for the material specification.
Quantity	Double	Specifies the quantity of the material specification.
Unit of Measure	String	Specifies the unit of measure of the material specification.

Inputs	Data Type	Description
Work Process Segment	DirectoryResource	Specifies the work process segment to add the material specification to.

AddOperationsMaterialBillItem (IProductionModel)

Adds a material specification item as an operations material bill item to the operations bill of materials (BOM) for the selected work definition.

Inputs	Data Type	Description
materialSpecificationName	String	Specifies the name of the material specification to add to the operations bill of material.
operationsMaterialBill	String	Specifies the name of the operations bill of materials (BOM) to add the material specification to.
operationsMaterialBillItemName	String	Specifies a name for the operations bill of materials (BOM) item.
workDefinitionAddress	DirectoryResource	Specifies the work definition to add the operations bill of materials (BOM) item to.
workSegmentAddress	DirectoryResource	Specifies the work process segment that the material specification is associated with.

Outputs	Description
OperationsMaterialBillItemAttributes	Returns the attributes of the operations bill of material (BOM) item.

AddPersonnelToProcessSegment (IProductionModel)

Adds a personnel specification to a work process segment.

Inputs	Data Type	Description
Description	String	Specifies a description for the personnel specification.
Name	String	Specifies a name for the personnel specification.
Personnel	DirectoryResource	Specifies the personnel resource to add as a specification to the selected work process segment.
Quantity	Double	Specifies the quantity of the personnel specification.
Unit of Measure	String	Specifies the unit of measure of the personnel specification.
Work Process Segment	DirectoryResource	Specifies the work process segment to add the personnel specification to.

AddSegmentToWorkDefinition (IProductionModel)

Associates a work definition segment with a work definition.

Inputs	Data Type	Description
Name	String	Specifies a display name for the association between the work definition segment and the work definition.
Order	Int	Specifies the execution order of an associated work definition segment.
Work Definition	DirectoryResource	Specifies the work definition to associate the work definition segment with.

Inputs	Data Type	Description
Work Definition Segment	DirectoryResource	Specifies the work definition segment to associate with the selected work definition.

AddWorkDataToProcessSegment (IProductionModel)

Adds a work data class to a process segment.

Inputs	Data Type	Description
Required	String	Indicates whether the work data for the selected work process segment will be automatically applied to segment responses. Valid values are: <ul style="list-style-type: none"> • False • True
Work Data Class	DirectoryResource	Specifies the work data class to add to the selected work process segment.
Work Process Segment	DirectoryResource	Specifies the work process segment to add the selected work data class to.

CreateOperationsMaterialBill (IProductionModel)

Creates an operations bill of materials (BOM) for the selected work definition.

Inputs	Data Type	Description
Description	DirectoryResource	Specifies the description of the new operations bill of materials (BOM).

Inputs	Data Type	Description
Name	String	Specifies the name of the new operations bill of materials (BOM).
workDefinitionAddress	String	Specifies the work definition to create the operations bill of materials (BOM) for.

Outputs	Description
OperationsMaterialBillAttributes	Returns the attributes of the operations bill of materials (BOM).

CreateProcessSegmentParameter (IProductionModel)

Adds a parameter to a process segment.

Inputs	Data Type	Description
DataType	String	Specifies the <i>data type</i> for the parameter.
Description	String	Specifies the description of the new process segment parameter.
Name	String	Specifies a display name for the parameter.
Unit of Measure	String	Specifies the unit of measure for the data type, such as gallons, inches, tons.
Value	String	Specifies the value for the selected data type. The options available depend on the selected data type.
Work Process Segment	DirectoryResource	Specifies the process segment you want to add a parameter to.

CreateWorkDataClass (IProductionModel)

Creates a work data class.

Inputs	Data Type	Description
Description	String	Specifies a description for the work data class.
Name	String	Specifies a display name for the work data class.

Outputs	Description
DirectoryResource	The address of the work data class.

CreateWorkDataClassStringProperty (IProductionModel)

Creates a work data class property.

Inputs	Data Type	Description
Description	String	Specifies the description of the work data class property.
Name	String	Specifies a display name for the property.
UoM (Unit of Measure)	String	Specifies the unit of measure for the property, such as gallons, inches, tons.
Value	String	Specifies the value for the property.
Work Data Class	DirectoryResource	Specifies the work data class to create a property for.

Outputs	Description	
ReturnValue		The LDAP address of the new string property.

CreateWorkDefinition (IProductionModel)

Creates a work definition.

Inputs	Data Type	Description
Description	String	Specifies a description for the work definition.
Name	String	Specifies a display name for the work definition.
Production Rule	String	Specifies an identification for a rule.
Version	String	Specifies the current version of the work definition.
Work Type	String	Specifies the work type that describes the category of work the work definition is associated with.

Outputs	Description	
DirectoryResource		The address of the new work definition.

CreateWorkDefinitionSegment (IProductionModel)

Creates a new work definition segment.

Inputs	Data Type	Description
Description	String	Specifies the description of the new work definition segment.

Inputs	Data Type	Description
Duration	Double	Specifies the work definition segment duration in minutes. The duration can be defined at any level of the process. Duration is not inherited through an association with another resource.
Name	String	Specifies a display name for the work definition segment.
Work Process Segment	DirectoryResource	Specifies the process segment you want to associate with the work definition segment.

Outputs	Description
DirectoryResource	The address of the new work definition segment.

CreateWorkSegmentParameter (IProductionModel)

Adds a parameter to a work definition segment.

Inputs	Data Type	Description
Data Type	String	Specifies the <i>data type</i> for the parameter.
Description	String	Specifies the description of the work definition segment parameter.
Name	String	Specifies a display name for the parameter.
Unit of Measure	String	Specifies the unit of measure for the data type, such as gallons, inches, tons.

Inputs	Data Type	Description
Value	String	Specifies the value for the selected data type. The options available depend on the selected data type.
Work Definition Segment	DirectoryResource	Specifies the work definition segment you want to add a parameter to.

DeleteExistingWorkDefinition (IProductionModel)

Deletes the specified work definition.

Inputs	Data Type	Description
Work Definition	DirectoryResource	Specifies the work definition to delete.

DeleteProcessSegmentParameter (IProductionModel)

Deletes the specified parameter from the selected work process segment.

Inputs	Data Type	Description
Name	String	Specifies the display name of the parameter you want to delete.
Work Process Segment	DirectoryResource	Specifies the work process segment that is the parent of the parameter you want to delete.

DeleteWorkDataClass (IProductionModel)

Deletes the selected work data class.

Inputs	Data Type	Description
Work Data Class	DirectoryResource	Specifies the work data class to delete.

DeleteWorkDataClassProperty (IProductionModel)

Deletes the specified property from the selected work data class.

Inputs	Data Type	Description
Name	String	Specifies the display name of the work data class property you want to delete.
Work Data Class	DirectoryResource	Specifies the work data class you are deleting the property from.

DeleteWorkDefinitionSegment (IProductionModel)

Deletes the selected work definition segment.



Note:

If the selected work definition segment is a parent to one or more child work definition segments, deleting it removes only the association with the child segments; that is, the child work definition segments are not deleted.

Inputs	Data Type	Description
Work Definition Segment	DirectoryResource	Specifies the work definition segment to delete.

DeleteWorkProcessSegment (IProductionModel)

Deletes the selected process segment.

Inputs	Data Type	Description
Work Process Segment	DirectoryResource	Specifies the work process segment to delete.

DeleteWorkSegmentParameter (IProductionModel)

Deletes the specified parameter from the selected work definition segment.

Inputs	Data Type	Description
Name	String	Specifies the display name of the parameter you want to delete..
Work Definition Segment	DirectoryResource	Specifies the work definition segment that is the parent of the parameter you want to delete.

GetMasterSegmentForWorkDefinition (IProductionModel)

Gets the master segment for a work definition.

Inputs	Data Type	Description
Work Definition	DirectoryResource	Specifies the work definition whose master segment you want to retrieve.

Outputs	Description
DirectoryResource	Returns the master segment for the specified work definition.

GetResourceForWorkDefinition (IProductionModel)

Gets the resource specification that is linked to a work definition.

Inputs	Data Type	Description
Work Definition	DirectoryResource	Specifies the work definition whose resource specification you want to retrieve.

Outputs	Description
DirectoryResource	Returns the resource specifications (that is, the equipment, material, and/or personnel specifica-

Outputs	Description	
		tions) for the specified work definition.

GetWorkDefinitionLinksForResource (IProductionModel)

Gets all of the work definitions linked to a resource specification.

Inputs	Data Type	Description
Linked Resource	DirectoryResource	Specifies the resource specification whose work definition(s) you want to retrieve.

Outputs	Description	
DirectoryResource[]		Returns all of the work definitions that the specified resource specification is linked with.

LinkResourceToWorkDefinition (IProductionModel)

Creates a resource link between a work definition and a resource specification.

Inputs	Data Type	Description
Linked Resource	DirectoryResource	Specifies the resource specification you want to link to a work definition.
Work Definition	DirectoryResource	Specifies the work definition you are linking a resource specification to.

ReadOperationsMaterialBill (IProductionModel)

Reads the attributes of an operations bill of material (BOM) associated with the selected work definition.

Inputs	Data Type	Description
operationsMaterialBillName	String	Specifies the name of the operations bill of materials (BOM) to read.
workDefinitionAddress	DirectoryResource	Specifies the work definition associated with the operations bill of material to be read.

Outputs	Description
OperationsMaterialBillItemAttributes	Returns attributes of the operations bill of material (BOM).

ReadSegmentParameters (IProductionModel)

Reads the parameters of the selected work definition segment.

Inputs	Data Type	Description
Work Definition Segment	DirectoryResource	Specifies the work definition segment that contains the parameters to be read.

Outputs	Description
ParameterCollection	Returns the attributes of the parameters contained by the selected work definition segment.

ReadWorkDefinition (IProductionModel)

Reads the attributes, associated segments, and the details of a work definition.

Inputs	Data Type	Description
Detail Level	WorkDefinitionDetailLevel	Select the level of work definition detail that you want the query to read:

Inputs	Data Type	Description
		<ul style="list-style-type: none"> • AttributesOnly • IncludeSegments
Segment Detail Level	SegmentDetailLevel	Specifies the detail level of the output. Valid values are: <ul style="list-style-type: none"> • AttributesOnly • IncludeParameters • IncludeEquipment • IncludeMaterial • IncludePersonnel • FullDetails
Work Definition	DirectoryResource	Specifies the work definition to be read.

Outputs	Description	
WorkDefinitionAttributes		A collection containing the work definition attributes and associated segments.

ReadWorkDefinitionAssociatedSegment (IProductionModel)

Reads the attributes of a work definition to work process segment association and the details of the associated work process segment.

Inputs	Data Type	Description
Detail Level	SegmentDetailLevel	Specifies the detail level of the output. Valid values are: <ul style="list-style-type: none"> • AttributesOnly • IncludeParameters • IncludeEquipment • IncludeMaterial • IncludePersonnel • FullDetails

Inputs	Data Type	Description
Name	String	Specifies the name of the associated work process segment to read.
Work Definition	DirectoryResource	Specifies the work definition that is associated with the work process segment you want to read.

Outputs	Description	
OrderedProductSegmentAttributes		Returns a collection of the details of the specified work process segment associated with the selected work definition.

ReadWorkDefinitionAssociatedSegmentsByParameter (IProductionModel)

Reads work definition segments that have a particular parameter ID and parameter value.

Inputs	Data Type	Description
Parameter Name	String	Specifies the name of the parameter you want to read.
Parameter Value	Object	Specifies the value of the parameter you want to read.
Work Definition	DirectoryResource	Specifies the work definition that is associated with the segments you want to read.

Outputs	Description	
ProductSegmentAttributes[]		Returns the work definition segments that contain the specific parameter name and parameter value that were requested.

ReadWorkDefinitionMaterialBills (IProductionModel)

Reads all of the operations bills of material (BOM) for the selected work definition.

Inputs	Data Type	Description
workDefinitionAddress	DirectoryResource	Specifies the work definition whose operations bills of material you want to read.

Outputs	Description
OperationsMaterialBillAttributes[]	Returns a collection of operations bills of material (BOM) attributes.

ReadWorkDefinitionSegment (IProductionModel)

Reads the attributes of the specified work definition segment.

Inputs	Data Type	Description
Detail Level	WorkDefinitionSegmentDetailLevel	Specifies the detail level of the output. Valid values are: <ul style="list-style-type: none"> • AttributesOnly • IncludeParameters • IncludeEquipment • IncludeMaterial • IncludePersonnel • ChildrenOnly • FullDetails
Work Definition Segment	DirectoryResource	Specifies the work definition segment you want to read.

Outputs	Description	
ProductSegmentAttributes		<p>Returns a collection of the details of the specified work definition segment.</p> <div data-bbox="1026 432 1414 741" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;"> <p> Note: If you select ChildrenOnly or FullDetails as the detail level, the immediate child segments are also returned.</p> </div>

RemoveChildWorkDefinitionSegment (IProductionModel)

Removes the association between a child work definition segment and its parent.

Inputs	Data Type	Description
Name	String	Specifies the name of the work definition segment association that you want to remove.
Parent	DirectoryResource	Specifies the parent work definition segment you are removing the child association from.

RemoveMasterSegmentFromWorkDefinition (IProductionModel)

Removes the master segment from a work definition.

Inputs	Data Type	Description
Work Definition	DirectoryResource	Specifies the work definition whose master segment you want to remove.

RemoveOperationsMaterialBill (IProductionModel)

Removes the specified operations bill of materials (BOM) from the selected work definition.

Inputs	Data Type	Description
operationsMaterialBillName	String	Specifies the name of the operations bill of materials (BOM) to remove.
workDefinitionAddress	DirectoryResource	Specifies the work definition to remove the operations bill of materials (BOM) from.

RemoveOperationsMaterialBillItem (IProductionModel)

Removes the specified item from the operations bill of materials (BOM) for the selected work definition.

Inputs	Data Type	Description
operationsMaterialBillItemName	String	Specifies the name of the operations bill of materials (BOM) item to remove.
operationsMaterialBillName	String	Specifies the name of the operations bill of materials (BOM) to remove the work definition from.
workDefinitionAddress	DirectoryResource	Specifies the work definition associated with the operations bill of materials (BOM) the item is being removed from.

RemoveResourceFromWorkDefinition (IProductionModel)

Removes the resource link to a resource specification for a work definition.

Inputs	Data Type	Description
Work Definition	DirectoryResource	Specifies the work definition whose resource link(s) you want to remove.

RemoveSegmentFromWorkDefinition (IProduction Model)

Removes the specified work definition segment association from a work definition.

Inputs	Data Type	Description
Name	String	Specifies the display name of the specification you want to remove from the selected work definition.
Work Definition	GUID	Specifies the work definition to remove the specified work definition segment from.

RemoveSpecificationFromProcessSegment (IProductionModel)

Removes a specification from a process segment.

Inputs	Data Type	Description
Name	String	Specifies the display name of the specification you want to remove from the selected work process segment.
Specification Type	ResourceTypes	Specifies the type of specification to remove from the selected work process segment. Valid values are: <ul style="list-style-type: none"> • Material • Equipment • Personnel
workSegmentAddress	DirectoryResource	Specifies the process segment you want to remove a specification from.

RemoveWorkDataFromProcessSegment (IProductionModel)

Removes a work data class from a process segment.

Inputs	Data Type	Description
Work Data Class	DirectoryResource	Specifies the work data class to remove from the selected work process segment.
Work Process Segment	DirectoryResource	Specifies the work process segment to remove the work data class from.

SetProcessSegmentMasterValue (IProductionModel)

Sets the value of the specified work process segment's master segment flag.

Inputs	Data Type	Description
IsMaster	Boolean	Indicates whether the specified work process segment is set as a master segment.
Work Process Segment	DirectoryResource	Specifies the work process segment to whose master segment flag you are setting.

SetWorkSegmentSpecification (IProductionModel)

Sets a resource for a class level specification for a work definition segment.

Inputs	Data Type	Description
Name	String	Specifies the display name of the specification you want to set for the selected work definition segment.
Specification	DirectoryResource	Specifies the specification to set for the selected work definition segment.
Specification Type	String	Specifies the specification type of the selected specification. Valid values are:

Inputs	Data Type	Description
		<ul style="list-style-type: none"> • Material • Equipment • Personnel
Work Definition Segment	DirectoryResource	Specifies the work definition segment to set the specification for.

SetWorkSegmentSpecificationQuantity (IProductionModel)

Sets the quantity for a specification associated with a work process segment.

Inputs	Data Type	Description
Name	String	Specifies the display name of the specification you want to set the quantity for.
Quantity	Int	Specifies the quantity of the specification associated with the selected work definition segment.
Specification Type	String	Specifies the specification type of the selected specification. Valid values are: <ul style="list-style-type: none"> • Material • Equipment • Personnel
Work Definition Segment	DirectoryResource	Specifies the work definition segment to set the specification quantity for.

UpdateChildWorkDefinitionSegmentAssociationName (IProductionModel)

Updates the name of the specified work definition segment association.

Inputs	Data Type	Description
New Name	String	Specifies the new name for the work definition segment association.
Old Name	String	Specifies the existing work definition segment association name you are changing.
Parent	DirectoryResource	Specifies the parent work definition segment whose unique association name you are changing.

UpdateWorkDefinitionSegmentOrder (IProductionModel)

Updates the order for a work segment association to a work definition.

Inputs	Data Type	Description
Name	String	Specifies the name of the work definition segment association.
Order	Double	Specifies the order in which the work definition segment will be processed.
Work Definition	DirectoryResource	Specifies the work definition you want to update.

UpdateWorkDefinitionDetails (IProductionModel)

Updates the details of a work definition.

Inputs	Data Type	Description
Description	String	Specifies a description for the work definition.
Name	String	Specifies a display name for the work definition.

Inputs	Data Type	Description
Production Rule	String	Specifies an identification for a rule.
Version	String	Specifies the current version of the work definition.
Work Definition	DirectoryResource	Specifies the work definition you want to update.
Work Type	String	Specifies the work type that describes the category of work the work definition is associated with. Valid values are: <ul style="list-style-type: none"> • Production • Maintenance • Quality • Inventory • Mixed • Unspecified

UpdateWorkDefinitionSegmentDetails (IProductionModel)

Updates the details of a work definition segment.

Inputs	Data Type	Description
Description	String	Specifies a description for the work definition segment.
Duration	Double	Specifies the work definition segment duration in minutes.
Name	String	Specifies a display name for the work definition segment.
Work Definition Segment	DirectoryResource	Specifies the work definition segment you want to update.

UpdateWorkProcessSegmentDetails (IProductionModel)

Updates a work process segment using information provided in attributes.

Inputs	Data Type	Description
Description	String	Specifies the description of the work process segment to update.
Duration	Double	Specifies the work process segment duration in minutes. The duration can be defined at any level of the process. Duration is not inherited through an association with another resource.
Location	DirectoryResource	Specifies the equipment resource associated with the work process segment.
Name	String	Specifies the display name of the work process segment whose details you want to update.
Unit of Measure	String	Specifies the unit of measure of the equipment specification associated with the work process segment.
Work Process Segment	DirectoryResource	Specifies the work process segment to be updated.
Work Type	String	Specifies the work type that describes the category of work the work definition is associated with.
Workflow	DirectoryResource	Specifies the workflow to associate with the work process segment.

This call method returns no data.

WriteSegmentParameters (IProductionModel)

Writes to the parameters of a specified work definition segment.

**Note:**

- This write modifies configuration data of the work definition segment, as well as all work requests that are subsequently created against any work definitions that contain the segment.
- In the Configure Method dialog box, click Next to select a work process segment whose parameters you want to write to. The parameters appear in the Inputs area of the activity configuration panel.

Inputs	Data Type	Description
Work Definition Segment input parameters	Dependent on each parameter's data type	Specifies the values for the parameters to write.
Work Definition Segment	DirectoryResource	Specifies a work definition segment that associated with the work process segment that contains the parameters you want to write to.

IProductionReport

IProductionReport allows you to read from the Production model.

On the activity configuration panel, you can use a *bind* or *expression* to define your input and output parameters.

Subtopics in this section describe each service method for this service interface.

QueryEquipmentActuals (IProductionReport)

Performs a query against the runtime database for equipment actuals associated with a given segment response based on the selected input parameters. All equipment actuals that match the specified query criteria are returned.

In the **Configure Method** dialog box, click **Next** to select query criteria to include as input parameters on the activity configuration panel.

**Note:**

If you do not select any query criteria as input parameters, your query returns all equipment actuals.

Criteria	Data Type	Description
Actual Equipment	DirectoryResource	<p>Specifies a resource to include in your query. Only actuals associated with that resource are returned by the query.</p> <p>You can specify an equipment instance or equipment class.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>
End Time	DateTime	<p>Specifies the utilization end date and time, in UTC, for the actual.</p> <p>As part of defining this parameter, you must also select the Criteria Type as either Not Set (to indicate an unspecified (that is, null) value) or Range (to indicate any value between and including two specified values).</p> <p>If you select Range as the criteria type, two fields are displayed in the Inputs section of the activity configuration panel: End Time (Min), to designate the beginning of the range, and End Time (Max), to designate the end of the range. Click the drop-down ar-</p>

Criteria	Data Type	Description
		<p>rows within the fields to open the calendar, where you can select the date and specify the time. Only actuals with end times within the specified range are returned by the query.</p>
Start Time	DateTime	<p>Specifies the utilization start date and time, in UTC, for the actual.</p> <p>As part of defining this parameter, you must also select the Criteria Type as either Not Set (to indicate an unspecified (that is, null) value) or Range (to indicate any value between and including two specified values).</p> <p>If you select Range as the criteria type, two fields are displayed in the Inputs section of the activity configuration panel: Start Time (Max), to designate the end of the range, and Start Time (Min), to designate the beginning of the range. Click the drop-down arrows within the fields to open the calendar, where you can select the date and specify the time. Only actuals with start times within the specified range are returned by the query.</p>
Segment Id	GUID	<p>Specifies a unique identifier for the segment response to include in your query. Only actuals associated with that segment response are returned by the query.</p>

Criteria	Data Type	Description
Use	String	Specifies the purpose for the actual resource to include in your query.

Outputs	Description
EquipmentActualAttributes[]	<p>Returns all of the actuals that match the query parameters.</p> <p>Use a <i>For Each Activity</i> to loop through all resources or use an <i>Index Activity</i> with an integer value. Indexing is supported only by the integer value; it does not support string values</p>

QueryMaterialActual (IProductionReport)

Performs a query against the runtime database for material actuals associated with a given segment response based on the selected input parameters. All material actuals that match the specified query criteria are returned.

In the **Configure Method** dialog box, click **Next** to select query criteria to include as input parameters on the activity configuration panel.



Note:

If you do not select any query criteria as input parameters, your query returns all material actuals.

Criteria	Data Type	Description
Actual Material	GUID	<p>Specifies a resource to include in your query. Only actuals associated with that resource are returned by the query.</p> <p>You can specify a material definition, material class, material lot, or material subplot.</p>

Criteria	Data Type	Description
		<div data-bbox="1024 268 1421 531" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: If you query by material lot or subplot, you must specify the ID of the resource. </div> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>
End Time	DateTime	<p>Specifies the utilization end date and time, in UTC, for the actual.</p> <p>As part of defining this parameter, you must also select the Criteria Type as either Not Set (to indicate an unspecified (that is, null) value) or Range (to indicate any value between and including two specified values).</p> <p>If you select Range as the criteria type, two fields are displayed in the Inputs section of the activity configuration panel: End Time (Min), to designate the beginning of the range, and End Time (Max), to designate the end of the range. Click the drop-down arrows within the fields to open the calendar, where you can select the date and specify the time. Only actuals with end times within the specified range are returned by the query.</p>

Criteria	Data Type	Description
Start Time	DateTime	<p>Specifies the utilization start date and time, in UTC, for the actual.</p> <p>As part of defining this parameter, you must also select the Criteria Type as either Not Set (to indicate an unspecified (that is, null) value) or Range (to indicate any value between and including two specified values).</p> <p>If you select Range as the criteria type, two fields are displayed in the Inputs section of the activity configuration panel: Start Time (Max), to designate the end of the range, and Start Time (Min), to designate the beginning of the range. Click the drop-down arrows within the fields to open the calendar, where you can select the date and specify the time. Only actuals with start times within the specified range are returned by the query.</p>
Segment Id	GUID	<p>Specifies a unique identifier for the segment response to include in your query. Only actuals associated with that segment response are returned by the query.</p>
Use	String	<p>Specifies the purpose for the actual resource to include in your query. The purpose you enter in this field is based on the resource actual uses defined by your business.</p>

Outputs	Description	
MaterialActualAttributes[]		<p>Returns all of the actuals that match the query parameters.</p> <p>Use a <i>For Each Activity</i> to loop through all resources or use an <i>Index Activity</i> with an integer value. Indexing is supported only by the integer value; it does not support string values.</p>

QueryPersonnelActual (IProductionReport)

Performs a query against the runtime database for personnel actuals associated with a given segment response based on the selected input parameters. All personnel actuals that match the specified query criteria are returned.

In the **Configure Method** dialog box, click **Next** to select query criteria to include as input parameters on the activity configuration panel.



Note:

If you do not select any query criteria as input parameters, your query returns all personnel actuals.

Criteria	Data Type	Description
Actual Personnel	DirectoryResource	<p>Specifies a resource to include in your query. Only actuals associated with that resource are returned by the query.</p> <p>You can specify a person (that is, a people resource) or a personnel class.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Uni-</p>

Criteria	Data Type	Description
		<p>versal Browser to select the resource you want to use.</p>
End Time	DateTime	<p>Specifies the utilization end date and time, in UTC, for the actual.</p> <p>As part of defining this parameter, you must also select the Criteria Type as either Not Set (to indicate an unspecified (that is, null) value) or Range (to indicate any value between and including two specified values).</p> <p>If you select Range as the criteria type, two fields are displayed in the Inputs section of the activity configuration panel: End Time (Min), to designate the beginning of the range, and End Time (Max), to designate the end of the range. Click the drop-down arrows within the fields to open the calendar, where you can select the date and specify the time. Only actuals with end times within the specified range are returned by the query.</p>
StartTime	DateTime	<p>Specifies the utilization start date and time, in UTC, for the actual.</p> <p>As part of defining this parameter, you must also select the Criteria Type as either Not Set (to indicate an unspecified (that is, null) value) or Range (to indicate</p>

Criteria	Data Type	Description
		<p>any value between and including two specified values).</p> <p>If you select Range as the criteria type, two fields are displayed in the Inputs section of the activity configuration panel: Start Time (Max), to designate the end of the range, and Start Time (Min), to designate the beginning of the range. Click the drop-down arrows within the fields to open the calendar, where you can select the date and specify the time. Only actuals with start times within the specified range are returned by the query.</p>
Segment Id	GUID	Specifies a unique identifier for the segment response to include in your query. Only actuals associated with that segment response are returned by the query.
Use	String	Specifies the purpose for the actual resource to include in your query. The purpose you enter in this field is based on the resource actual uses defined by your business.

Outputs	Description	
PersonnelActualAttributes[]		<p>Returns all of the actuals that match the query parameters.</p> <p>Use a <i>For Each Activity</i> to loop through all resources or use an <i>In-</i></p>

Outputs	Description	
		<i>dex Activity</i> with an integer value. Indexing is supported only by the integer value; it does not support string values.

QuerySegmentResponses (IProductionReport)

Performs a query against the runtime database for segment responses based on the selected input parameters. All segment responses that match the specified query criteria are returned.

In the **Configure Method** dialog box, click **Next** to select query criteria to include as input parameters on the activity configuration panel.



Note:

If you do not select any query criteria as input parameters, your query returns all segment responses.

Criteria	Data Type	Description
Actual Equipment	DirectoryResource	<p>Specifies the actual resource to include in your query. Only segment responses that are associated with this resource are returned by the query.</p> <p>You can specify an equipment instance or equipment class.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>
Actual Equipment End Time	DateTime	<p>Specifies the utilization end date and time, in UTC, for the actual.</p> <p>As part of defining this parameter, you must also select the Cri-</p>

Criteria	Data Type	Description
		<p>Criteria Type as either Not Set (to indicate an unspecified (that is, null) value) or Range (to indicate any value between and including two specified values).</p> <p>If you select Range as the criteria type, two fields are displayed in the Inputs section of the activity configuration panel: one to designate the beginning (that is, <i>Min</i>) of the actual time range, and one to designate the end (that is, <i>Max</i>) of the actual time range. Click the drop-down arrows within the fields to open a calendar tool, where you can select the date and specify the time. Only responses with actuals that are of the designated type and that have end times within the specified range are returned by the query.</p>
Actual Equipment Start Time	DateTime	<p>Specifies the utilization start date and time, in UTC, for the actual.</p> <p>As part of defining this parameter, you must also select the Criteria Type as either Not Set (to indicate an unspecified (that is, null) value) or Range (to indicate any value between and including two specified values).</p> <p>If you select Range as the criteria type, two fields are displayed in the <i>Inputs</i> section of the ac-</p>

Criteria	Data Type	Description
		<p>activity configuration panel: one to designate the beginning (that is, <i>Min</i>) of the actual time range, and one to designate the end (that is, <i>Max</i>) of the actual time range. Click the drop-down arrows within the fields to open a calendar tool, where you can select the date and specify the time. Only responses with actuals that are of the designated type and that have start times within the specified range are returned by the query.</p>
Actual Material	GUID	<p>Specifies the actual resource to include in your query. Only segment responses that are associated with this resource are returned by the query.</p> <p>You can specify a material definition, material class, material lot, or material subplot.</p> <div data-bbox="1024 1325 1419 1591" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;"> <p> Note: If you query by material lot or subplot, you must specify the ID of the resource.</p> </div> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>

Criteria	Data Type	Description
Actual Material End Time	DateTime	<p>Specifies the utilization end date and time, in UTC, for the actual.</p> <p>As part of defining this parameter, you must also select the Criteria Type as either Not Set (to indicate an unspecified (that is, null) value) or Range (to indicate any value between and including two specified values).</p> <p>If you select Range as the criteria type, two fields are displayed in the Inputs section of the activity configuration panel: one to designate the beginning (that is, <i>Min</i>) of the actual time range, and one to designate the end (that is, <i>Max</i>) of the actual time range. Click the drop-down arrows within the fields to open a calendar tool, where you can select the date and specify the time. Only responses with actuals that are of the designated type and that have end times within the specified range are returned by the query.</p>
Actual Material Start Time	DateTime	<p>Specifies the utilization start date and time, in UTC, for the actual.</p> <p>As part of defining this parameter, you must also select the Criteria Type as either Not Set (to indicate an unspecified (that is, null) value) or Range (to indicate</p>

Criteria	Data Type	Description
		<p>any value between and including two specified values).</p> <p>If you select Range as the criteria type, two fields are displayed in the Inputs section of the activity configuration panel: one to designate the beginning (that is, <i>Min</i>) of the actual time range, and one to designate the end (that is, <i>Max</i>) of the actual time range. Click the drop-down arrows within the fields to open a calendar tool, where you can select the date and specify the time. Only responses with actuals that are of the designated type and that have start times within the specified range are returned by the query.</p>
Actual Personnel	DirectoryResource	<p>Specifies the actual resource to include in your query. Only segment responses that are associated with this resource are returned by the query.</p> <p>You can specify a person (that is, a people resource) or a personnel class.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>

Criteria	Data Type	Description
Actual Personnel End Time	DateTime	<p>Specifies the utilization end date and time, in UTC, for the actual.</p> <p>As part of defining this parameter, you must also select the Criteria Type as either Not Set (to indicate an unspecified (that is, null) value) or Range (to indicate any value between and including two specified values).</p> <p>If you select Range as the criteria type, two fields are displayed in the Inputs section of the activity configuration panel: one to designate the beginning (that is, <i>Min</i>) of the actual time range, and one to designate the end (that is, <i>Max</i>) of the actual time range. Click the drop-down arrows within the fields to open a calendar tool, where you can select the date and specify the time. Only responses with actuals that are of the designated type and that have end times within the specified range are returned by the query.</p>
Actual Personnel Start Time	DateTime	<p>Specifies the utilization start date and time, in UTC, for the actual.</p> <p>As part of defining this parameter, you must also select the Criteria Type as either Not Set (to indicate an unspecified (that is, null) value) or Range (to indicate</p>

Criteria	Data Type	Description
		<p>any value between and including two specified values).</p> <p>If you select Range as the criteria type, two fields are displayed in the Inputs section of the activity configuration panel: one to designate the beginning (that is, <i>Min</i>) of the actual time range, and one to designate the end (that is, <i>Max</i>) of the actual time range. Click the drop-down arrows within the fields to open a calendar tool, where you can select the date and specify the time. Only responses with actuals that are of the designated type and that have start times within the specified range are returned by the query.</p>
Detail Level	SegmentResponseDetailLevel	<p>Specifies the detail level of the output. Valid values are:</p> <ul style="list-style-type: none"> • AttributesOnly • IncludeWorkData • IncludeEquipmentActuals • IncludeMaterialActuals • IncludePersonnelActuals • ChildrenOnly • FullDetails <div data-bbox="1024 1640 1421 1837" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: This is a mandatory criterion displayed by default in the Inputs section of</p> </div>

Criteria	Data Type	Description
		 the activity configuration panel.
End Time	DateTime	<p>Specifies the segment response end time.</p> <p>If you select this parameter, two fields are displayed in the Inputs section of the activity configuration panel: one to designate the beginning (that is, <i>Min</i>) of the response time range, and one to designate the end (that is, <i>Max</i>) of the response time range. Click the drop-down arrows within the fields to open a calendar tool, where you can select the date and specify the time. Only responses that have end times within the specified range are returned by the query.</p>
Location	DirectoryResource	<p>Specifies the equipment location to include in your query.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>
Parent Segment Response	GUID	Specifies the ID of the parent segment response to include in your query.
S95 ID	String	Specifies the name of the segment response to include in your query.

Criteria	Data Type	Description
Start Time	DateTime	<p>Specifies the segment response start time.</p> <p>If you select this parameter, two fields are displayed in the Inputs section of the activity configuration panel: one to designate the beginning (that is, <i>Min</i>) of the response time range, and one to designate the end (that is, <i>Max</i>) of the response time range. Click the drop-down arrows within the fields to open a calendar tool, where you can select the date and specify the time. Only responses that have start times within the specified range are returned by the query.</p>
Work Data Class	DirectoryResource	<p>Specifies the work data class to include in your query.</p> <div data-bbox="1024 1199 1419 1556" style="border: 1px solid #00a0c0; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note: If you select a work data class in the Configure Method dialog box, you can also configure parameters based on the properties of that class.</p> </div> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>

Criteria	Data Type	Description
Work Process Segment	DirectoryResource	Specifies the work process segment to include in your query. In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.
Work Request	DirectoryResource	Specifies the work request to include in your query. In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.
Work Response	GUID	Specifies the ID of the work response to include in your query.

Outputs	Description
SegmentResponseAttributes[]	Returns all of the segment responses that match the query parameters. Use a <i>For Each Activity</i> to loop through all resources or use an <i>Index Activity</i> with an integer value. Indexing is supported only by the integer value; it does not support string values.

QueryWorkRequests (IProductionReport)

Performs a query against the runtime database for work requests based on the selected input parameters.

In the **Configure Method** dialog box, click **Next** to select query criteria to include as input parameters on the activity configuration panel.

Inputs	Data Type	Description
Creation Time (Max)	DateTime	Specifies the latest creation date and time to include in your query. Only responses that occurred on or before this date/time are returned.
Creation Time (Min)	DateTime	Specifies the earliest creation date and time to include in your query. Only responses that occurred on or after this date/time are returned.
End Time (Max)	DateTime	<p>Specifies the latest end date and time to include in your query. Only responses that occurred on or before this date/time are returned.</p> <p>In the Configure Method dialog box, click Next to specify the criteria type to use in your query.</p> <div data-bbox="1024 1283 1419 1864" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: If you select Range, then the End Time (Max) and End Time (Min) parameters are available and all responses matching the parameter inputs are returned. If you select Not Set, then the query returns only those responses that do not have an end date/time defined.</p> </div>

Inputs	Data Type	Description
		<p>In the Inputs section of the activity configuration panel, click the drop-down arrow to set the date and time.</p>
End Time (Min)	DateTime	<p>Specifies the earliest end date and time to include in your query. Only responses that occurred on or after this date/time are returned.</p> <p>In the Configure Method dialog box, click Next to specify the criteria type to use in your query.</p> <div data-bbox="1024 873 1419 1457" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: If you select Range, then the End Time (Max) and End Time (Min) parameters are available and all responses matching the parameter inputs are returned. If you select Not Set, then the query returns only those responses that do not have an end date/time defined.</p> </div> <p>In the Inputs section of the activity configuration panel, click the drop-down arrow to set the date and time.</p>
Location	DirectoryResource	<p>Specifies the equipment location to include in your query.</p> <p>In the Inputs section of the activity configuration panel, click the</p>

Inputs	Data Type	Description
		ellipsis button to open the Universal Browser to select the resource you want to use.
Priority (Max)	Int	Specifies the highest priority value to include in your query. Only responses with a priority value equal to or less than this value are returned.
Priority (Min)	Int	Specifies the lowest priority value to include in your query. Only responses with a priority value equal to or greater than this value are returned.
Start Time (Max)	DateTime	<p>Specifies the latest start date and time to include in your query. Only responses that occurred on or before this date/time are returned.</p> <p>In the Configure Method dialog box, click Next to specify the criteria type to use in your query.</p> <div data-bbox="1024 1314 1419 1892" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note:</p> <p>If you select Range, then the Start Time (Max) and Start Time (Min) parameters are available and all responses matching the parameter inputs are returned. If you select Not Set, then the query returns only those responses that do not have a start date/time defined.</p> </div>

Inputs	Data Type	Description
		<p>In the Inputs section of the activity configuration panel, click the drop-down arrow to set the date and time.</p>
Start Time (Min)	DateTime	<p>Specifies the earliest start date and time to include in your query. Only responses that occurred on or after this date/time are returned.</p> <p>In the Configure Method dialog box, click Next to specify the criteria type to use in your query.</p> <div data-bbox="1024 873 1419 1457" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note:</p> <p>If you select Range, then the Start Time (Max) and Start Time (Min) parameters are available and all responses matching the parameter inputs are returned. If you select Not Set, then the query returns only those responses that do not have a start date/time defined.</p> </div> <p>In the Inputs section of the activity configuration panel, click the drop-down arrow to set the date and time.</p>
Status	String	<p>Specifies the work request status to use in your query. Valid values are:</p>

Inputs	Data Type	Description
		<ul style="list-style-type: none"> • Draft • Unscheduled • Pending • Running • Paused • Completed • Failed • Canceled • Unknown <p>In the Configure Method dialog box, click Next to select a status. You cannot select or enter a status on the activity configuration panel.</p>
Work Definition	DirectoryResource	<p>Specifies the work definition to include in your query.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>

Outputs	Description	
WorkRequestCollection		<p>Returns all of the work requests that match the query parameters.</p> <p>Use a <i>For Each Activity</i> to loop through all resources, an <i>Index Activity</i> with an integer value, or the name of the work request.</p>

**Important:**

The returned collection of data is based on user-defined values and is dependent on your published data. For information on the attributes that can be read from, see *Work Requests*.

QueryWorkResponses (IProductionReport)

Performs a query against the runtime database for work responses based on the selected input parameters. All work responses that match the specified query criteria are returned.

In the **Configure Method** dialog box, click **Next** to select query criteria to include as input parameters on the activity configuration panel.

**Note:**

If you do not select any query criteria as input parameters, your query returns all work responses.

Inputs	Data Type	Description
Detail Level	WorkResponseDetailLevel	<p>Specifies the detail level of the output. Select one of the following:</p> <ul style="list-style-type: none"> • IncludeSegmentResponses • AttributesOnly
End Time (Max)	DateTime	<p>Specifies the latest end date and time to include in your query. Only responses that occurred on or before this date/time are returned.</p> <p>In the Configure Method dialog box, click Next to specify the criteria type to use in your query.</p> <div data-bbox="1032 1692 1083 1740" data-label="Image"></div> <p>Note:</p> <p>If you select Range, then the End Time (Max) and End Time (Min) parameters</p>

Inputs	Data Type	Description
		<p> ters are available and all responses matching the parameter inputs are returned. If you select Not Set, then the query returns only those responses that do not have an end date/time defined.</p> <p>In the Inputs section of the activity configuration panel, click the drop-down arrow to set the date and time.</p>
End Time (Min)	DateTime	<p>Specifies the earliest end date and time to include in your query. Only responses that occurred on or after this date/time are returned.</p> <p>In the Configure Method dialog box, click Next to specify the criteria type to use in your query.</p> <p> Note: If you select Range, then the End Time (Max) and End Time (Min) parameters are available and all responses matching the parameter inputs are returned. If you select Not Set, then the query returns only those responses that do not have an end date/time defined.</p>

Inputs	Data Type	Description
		In the Inputs section of the activity configuration panel, click the drop-down arrow to set the date and time.
Location	DirectoryResource	<p>Specifies the equipment location to include in your query.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>
Origin Work Request	DirectoryResource	<p>Specifies the originating work request to include in your query.</p> <div data-bbox="1024 936 1419 1293" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin: 10px 0;"> <p> Note: This parameter applies only if any of the work responses you are querying have been moved from one work request to another.</p> </div> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>
Response State	String	<p>Specifies the work response state to include in your query.</p> <ul style="list-style-type: none"> • Ready • Completed • Holding • Aborted

Inputs	Data Type	Description
Segment Dtail Level	SegmentResponseDetailLevel	<p>Specifies the detail level of the output. Valid values are:</p> <ul style="list-style-type: none"> • AttributesOnly • IncludeWorkData • IncludeEquipmentActuals • IncludeMaterialActuals • IncludePersonnelActuals • ChildrenOnly • FullDetails
Start Time (Max)	DateTime	<p>Specifies the earliest start date and time to include in your query. Only responses that occurred on or after this date/time are returned.</p> <p>In the Configure Method dialog box, click Next to specify the criteria type to use in your query.</p> <div data-bbox="1027 1142 1421 1724" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note:</p> <p>If you select Range, then the Start Time (Max) and Start Time (Min) parameters are available and all responses matching the parameter inputs are returned. If you select Not Set, then the query returns only those responses that do not have a start date/time defined.</p> </div> <p>In the Inputs section of the activity configuration panel, click the</p>

Inputs	Data Type	Description
Start Time (Min)	DateTime	<p>drop-down arrow to set the date and time.</p> <p>Specifies the earliest start date and time to include in your query. Only responses that occurred on or after this date/time are returned.</p> <p>In the Configure Method dialog box, click Next to specify the criteria type to use in your query.</p> <div data-bbox="1027 783 1419 1367" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: If you select Range, then the Start Time (Max) and Start Time (Min) parameters are available and all responses matching the parameter inputs are returned. If you select Not Set, then the query returns only those responses that do not have a start date/time defined.</p> </div> <p>In the Inputs section of the activity configuration panel, click the drop-down arrow to set the date and time.</p>
Work Request	DirectoryResource	<p>Specifies the work request to include in your query.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Uni-</p>

Inputs	Data Type	Description
		versal Browser to select the resource you want to use.

Outputs	Description
WorkResponseCollection	Returns all of the work responses that match the query parameters. Use a <i>For Each Activity</i> to loop through all resources, an <i>Index Activity</i> with an integer value, or the name of the work response.



Important:

The returned collection of data is based on user-defined values and is dependent on your published data. For information on the attributes that can be read from, see *Work Requests*.

IProductionRuntime

The IProductionRuntime service interface relates to production oversight.

You can use the IProductionRuntime service interface call methods in workflows for scenarios commonly encountered in overseeing production, including creating, modifying, querying, and deleting work requests, work responses, segment responses, actuals, and other production-related items.

Related reference

[Production Runtime Call Methods \(on page 1933\)](#)

Related information

[Common Production Runtime Workflow Scenarios \(on page 1906\)](#)

Common Production Runtime Workflow Scenarios

Common production runtime scenarios center around creating, modifying, querying, and deleting work requests, work responses, segment responses, actuals, and other production-related items.

The IProductionRuntime service interface provides call methods that can be used in workflows for scenarios commonly encountered in overseeing production.

Create a work request using a workflow

This procedure describes how to create and configure a work request using a workflow.

About this task

You can create expressions or bind properties for many of these properties, depending on your workflow requirements.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, add a resource to the workflow model.
3. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow. See **Add a resource to the workflow model**.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. Insert an activity:
 - At the workflow level:
 - From the Standard Activities panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the Standard Activities panel, click and drag the activity you want to insert to a drop point.
7. On the activity configuration panel, click **Configure Method**.
The Configure Method dialog box appears, displaying a list of the available call method service providers.
 8. Double-click **IProductionRuntime** to display the list of available call methods.
 9. Click **CreateWorkRequest**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
 10. Optionally:
 - a. In the **Description** field, enter a description or expression, or bind a property or parameter to the method.
 - b. In the **S95 ID** field, enter a name or expression, or bind a property or parameter to the method.

11. In the **Work Definition** field, click the ellipsis button, and then select a work definition.
12. Click **Save**.
13. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

Create a work segment response for a work request using a workflow

Before you begin

In a workflow subprocess, you use two call method activities to perform this task. You can create expressions or bind properties for many of these properties, depending on your workflow requirements.

Procedure

1. In the navigator, click **Workflow > Workflows**.
 2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow. See **Add a Resource to the Workflow Model**.
 3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
 4. Click the **Workflow Editor** tab.
 5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
-  **Note:**
Activities can exist only between the presubprocess and postsubprocess.
- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
 6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IProductionRuntime** to display the list of available call methods.
 8. Click **CreateWorkResponse**.
 9.
 - a. Click **Next** to select a work type, and then click **Finish**.
The input parameters appear in the activity configuration panel.

- b. In the **Description** field, enter a description or expression, or bind a property or parameter to the method.
 - c. In the **Location** field, click the ellipsis button, and then select an equipment location.
 - d. In the **S95 ID** field, enter a name or expression, or bind a property or parameter to the method.
10. In the **Work Request** field, click the ellipsis button.
The **Universal Browser** appears.
11. Select the work request you want to associate with the work response.
12. Repeat steps 5-7.
13. Click **CreateSegmentResponse**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
14. Optionally:
 - a. In the **Description** field, enter a description or expression, or bind a property or parameter to the method.
 - b. In the **End Time** and **Start Time** fields, select a date and time, or add an expression, or bind a property or parameter to the method.
 - c. In the **Location** field, click the ellipsis button, and then select an equipment location.
 - d. In the **S95 ID** field, enter a name or expression, or bind a property or parameter to the method.
15. In the **Work Process Segment** field, click the ellipsis button, and then select a process segment to associate with this segment response.
16. In the **Parent ID** field, click the arrow, and then select **Bind**.
The **Select Variable** dialog box appears.
17. Expand the CreateWorkResponse call method added in step 8, select a ReturnValue property, and then click **OK**.
18. Click **Save**.
19. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.
When the workflow runs, the segment response is created and can be viewed in the Response Information tab of the work request associated with the work request selected in step 11.

Write work data for a work segment response using a workflow

You can create expressions or bind properties for many of these properties, depending on your workflow requirements.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IProductionRuntime** to display the list of available call methods.



Note:

If you selected the **Required Class** check box on the **Work Data** tab of the associated work process segment, then go to step 8. If not, go to step 9.

8. Click **AddWorkDataClassToResponse**, and then click **OK**.
 - a. In the **Segment Response ID** field, click the arrow, and then select **Bind**.
The **Select Variable** dialog box appears.
 - b. From the **Activities** tab, select a **ReturnValue** for the segment response, and then click **OK**.
 - c. If required, in **Work Data Class**, bind a property the resource.
9. Click **WriteWorkData**.

- a. If required, to configure the data properties to write, click **Next**.
 - b. Click **Add classes of work data properties**, and then select a work data class.
The properties for the selected work data class appear.
 - c. To change properties, click **Remove classes of work data properties**, and then select a different work data class.
 - d. Click **Select all** or select the check box of a specific property, and then click **Finish**. Each property you select appears as an input parameter on the activity configuration panel.
 - e. For each property, enter a new value or expression, or bind a property or parameter to the method.
 - f. In the **Segment Response ID** field, click the arrow, and then select **Bind**.
The **Select Variable** dialog box appears.
 - g. From the **Activities** tab, select a **ReturnValue** for the segment response, and then click **OK**.
10. Click **Save**.
 11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

Create a segment requirement

You can create expressions or bind properties for many of these properties, depending on your workflow requirements.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:

- At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IProductionRuntime** to display the list of available call methods.
 8. Click **CreateSegmentRequirement**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
 9. In the **Name** field, enter a name for the segment requirement, or bind a property or parameter to the method.
 10. In the **Order** field, enter an execution order for the segment requirement, or bind a property or parameter to the method.
 11. **Optional:** In the **Parent Requirement ID** field, enter the ID of the parent segment requirement that the segment requirement is associated with.
 12. In the **Work Definition Segment** field, click the ellipsis button, and then select a work definition segment.
The segment requirement uses this definition to define properties and hierarchy.
 13. In the **Work Request** field, click the ellipsis button, and then select a work request.
The segment requirement will be associated with this work request.
 14. Click **Save**.
 15. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

Set a lot or subplot for a material requirement

The following procedure is an example of how to create a workflow that specifies a material lot or subplot for a material requirement that is part of a work request.

Before you begin

Before you begin, you must have the appropriate equipment and material resources (including lots and sublots) available. In addition, you must have your production process configured; that is, work process segments, work definition segments, and work definitions must all exist. You can use an existing work request, or you can create a new work request. This procedure includes steps for creating a work request as part of the workflow.

Procedure

1. Add a workflow that will be used to set a material lot or subplot for a material requirement.



Note:

Activities can be inserted at both the workflow and subprocess levels.

2. Insert an activity:

- At the workflow level:

- From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:

- Double-click the local subprocess you want to add the activity to.
- From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

3. On the activity configuration panel, click **Configure Method**.

The **Configure Method** dialog box appears, displaying a list of the available call method service providers.

4. Double-click **IProductionRuntime** to display the list of available call methods.

5. Click **CreateWorkRequest**, and then click **Finish**.

The input parameters appear in the activity configuration panel.

6. Optionally:

- a. In the **Description** field, enter a description or expression, or bind a property or parameter to the method.
- b. In the **S95 ID** field, enter a name or expression, or bind a property or parameter to the method.

7. In the **Work Definition** field, click the ellipsis button, and then select a work definition.

8. Click **Save**.
 9. Insert another Call Method activity after the CreateWorkRequest call method.
 10. On the activity configuration panel, click **Configure Method**.

The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 11. Double-click **IProductionRuntime** to display the list of available call methods.
 12. Click **ReadSegmentRequirements**, and then click **Finish**.

The input parameters appear in the activity configuration panel.
 13. Configure the call method as follows:
 - a. From the **Detail Level** list, select **IncludeMaterialRequirements**.
 - b. In the **Work Request** field, click the arrow, and then select **Bind**.

The **Select Variable** dialog box appears.
 - c. Expand the **CreateWorkRequest** call method you created in step 3, select the **ReturnValue** parameter, and then click **OK**.
 14. Insert an Index activity after the ReadSegmentRequirements call method and configure it to read from the collection returned by the ReadSegmentRequirements call method.
 15. Insert a ForEach activity after the Index activity, and then configure it as follows:
 - a. In the **Collection** field, click the arrow, and then select **Bind**.

The **Select Variable** dialog box appears.
 - b. Expand the Index activity, select the **MaterialRequirements** parameter, and then click **OK**.
 16. Insert a Call Method activity inside the ForEach activity, and then configure it as follows:
 - a. On the activity configuration panel, click **Configure Method**.

The Configure Method dialog box appears, displaying a list of the available call method service providers.
 - b. Double-click **IMaterial**, click **QueryMaterialLots**, and then click **Next**.
-  **Tip:**
If you are setting a material subplot, then use the **QueryMaterialSublots** call method and the following sub-steps.
- c. From the list of criteria, select **Material Definition**, and then click **Finish**.

- d. On the activity configuration panel, from the **Detail Level** list, select **OnlyAttributes**.
- e. In the **Material Definition** field, click the ellipsis button, select the material definition that is associated with the material lot or subplot you want to set, and then click **OK**.

**Note:**

The material definition you select must be consistent with the material requirement; that is, it must be the same as either the material definition or material class set in the requirement.

17. Insert an Index activity inside the ForEach activity, below the QueryMaterialLots (or QueryMaterialSublots) call method, and configure it to read from the collection returned by the call method.
18. Insert a Call Method activity inside the ForEach activity, below the Index activity, and then configure it as follows.
 - a. On the activity configuration panel, click **Configure Method**
 - b. Expand **IProductionRuntime**, click **SetMaterialRequirementLotOrSublot**, and then click **Finish**.
 - c. On the activity configuration panel, in the **Lot or Sublot Id** field, click the arrow. The **Select Variable** dialog box appears.
 - d. Expand the Index activity you configured in step 14, select the **Id** parameter, and then click **OK**.
 - e. From the **Lot Type** list, select **MaterialLot** or **MaterialSublot**, depending on whether you are setting a material lot or subplot for the material requirement.
 - f. In the **Requirement Id** field, click the arrow. The **Select Variable** dialog box appears.
 - g. Expand the ForEach activity you configured in step 15, select the **Id** parameter, and then click **OK**.
19. Click **Save**.
20. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

What to do next



Note:

If you perform a `ReadSegmentRequirements` call method on this work request after running this workflow, it will return the lots and/or sublots that you set on the work request's material requirements.

Create a resource actual

This procedure describes all of the steps required to add and configure resource actuals for a specified segment response.

Before you begin

Depending on the type of actual you are creating, before you proceed with this procedure, make sure you have created the appropriate equipment, material, or personnel resource you are configuring the actual(s) for.

About this task

You can create expressions or bind properties for many of these properties, depending on your workflow requirements.



Attention:

The following procedure is provided as an example only and is not meant to be used in a production environment.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. Click **Add**, and then add and configure a workflow as follows:
 - a. On the workflow configuration panel, in the **Personnel** field, click the ellipsis button, and then select the personnel resource you want to assign to this workflow.
 - b. Click **Edit** next to **Parameters**.
The **Edit Parameters** dialog box appears.
 - c. Click **Add**, and then configure an input parameter with a **Type** of `String`, a **Direction** of `In`, and a unique name to bind to the **S95 Id** of the work request you will create in step 5b.

- d. Click **Add**, and then configure an output parameter to represent the resource actual's display name, with a **Type** of `String`, a **Direction** of `Out`, and a unique name to bind to the Write activity you will add in step 20d.
 - e. Click **Add**, and then configure an output parameter to represent the results of the workflow, with a **Type** of `Boolean`, a **Direction** of `Out`, and a unique name to bind to the Write activity you will add in step 20h.
 - f. Click **OK**.
 - g. Add a local subprocess to the workflow, click **Edit** on the subprocess configuration panel, enter a unique name to represent the resource actual configuration, and then click **OK**.
 - h. Add another local subprocess after the first, click **Edit** on the subprocess configuration panel, enter a unique name to represent the query you will configure in step 9, and then click **OK**.
3. Double-click the resource actual subprocess, and then insert a Call Method activity.
 4. On the activity configuration panel, click **Configure Method**.

The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 5. In the **IProductionRuntime** methods, click **CreateWorkRequest**, click **Finish**, and then configure the input parameters.
 - a. **Optional:** In the **Description** field, enter information to uniquely describe the work request.
 - b. In the **S95 Id** field, bind to the input parameter you configured for the workflow in step 2c.
 - c. In the **Work Definition** field, click the ellipsis button, and then select the work definition to associate with the work request.
 6. Insert another Call Method activity, and then click **Configure Method**.
 7. In the **IProductionRuntime** methods, click **CreateWorkResponse**, and then click **Next**.
 - a. In the **Item specific configuration** area, select a work type from the drop-down list, and then click **Finish**.
 - b. On the activity configuration panel, click **Edit**, enter a unique name to represent the call method, and then click **OK**.
 - c. **Optional:** In the **Description** field, enter information to uniquely describe the work response.
 - d. In the **Location** field, click the ellipsis button, and then select the equipment location to associate with the work response.

- e. In the **S95 Id** field, enter a name for the work response, or click the arrow and create an expression; for example, "`<workresponsename>`" + `ToString(CurrentDateTime())`, which will create a unique name for every work response that is created when this workflow is run. The value of "`<workresponsename>`" is the name you choose to give to the work responses.
 - f. In the **Work Request** field, click the arrow, and then bind the work response to the work request you created in step 5.
8. Insert another Call Method activity, and then click **Configure Method**.
9. In the **IProductionRuntime** methods, select **CreateSegmentResponse**, and then click **Finish**.
 - a. On the activity configuration panel, click **Edit**, enter a unique name to represent the call method, and then click **OK**.
 - b. **Optional:** In the **Description** field, enter information to uniquely describe the segment response.
 - c. **Optional:** In the **End Time** field, click the arrow, and then select the terminating date and time for the segment response.
 - d. In the **Location** field, click the ellipsis button, and then select the equipment location to associate with the segment response.
 - e. In the **Parent Id** field, click the arrow, and then bind the segment response to the output parameter of the work response you created in step 6.
 - f. In the **S95 Id** field, enter a name for the segment response, or click the arrow and create an expression; for example, "`<segmentresponsename>`" + `ToString(CurrentDateTime())`, which will create a unique name for every segment response that is created when this workflow is run. The value of "`<workresponsename>`" is the name you choose to give to the segment responses.
 - g. **Optional:** In the **Start Time** field, click the arrow, and then select the starting date and time for the segment response.
 - h. In the **Work Process Segment** field, click the ellipsis button, and then select the work process segment to associate with the segment response.
10. Insert another Call Method activity, and then click **Configure Method**.

11. In the **IProductionRuntime** methods, select the method applicable to the type of actual you are creating, based on your need to set initial start and end time values:

If you...	Then select one of the following
Need to set initial start and end time values	<ul style="list-style-type: none"> ◦ CreateEquipmentActualEx ◦ CreateMaterialActualEx ◦ CreatePersonnelActualEx
Do not need to set initial start and end time values	<ul style="list-style-type: none"> ◦ CreateEquipmentActual ◦ CreateMaterialActual ◦ CreatePersonnelActual

12. Click **Finish**.

13. Configure the call method as follows:

- a. On the activity configuration panel, click **Edit**, enter a unique name to represent the call method, and then click **OK**.
- b. **Optional:** In the **Description** field, enter information to uniquely describe the resource actual you want to create.
- c. If the call method you are configuring allows you to designate the initial **End Time** value, click the drop-down arrow to set the date and time.
- d. Depending on the type of resource actual you are creating, in the **Equipment**, **Material**, or **Personnel** field, click the ellipsis button, select the appropriate resource for the actual, and then click **OK**.
- e. **Optional:** If you are creating a material resource actual, then, in the **Lot or Sublot ID** field, enter a literal value, or click the drop-down arrow to perform a bind or create an expression.



Note:

The data type of this field must be GUID.

- f. In the **Quantity** field, enter a literal value, or click the drop-down arrow to perform a bind or create an expression.
- g. **Optional:** In the **Requirement ID** field, enter the resource requirement you want to associate with the actual, or click the drop-down arrow to perform a bind or create an expression.

- a. On the activity configuration panel, click **Edit**, enter a unique name to represent the activity, and then click **OK**.
 - b. In the **Collection** field, click the arrow, bind to the return value of the call method you added in step 9, and then click **OK**.
20. Insert another For Each activity after the first For Each activity, and then configure it as follows:
- a. On the activity configuration panel, click **Edit**, enter a unique name to represent the activity, and then click **OK**.
 - b. In the **Collection** field, click the arrow, bind to the resource actual property of the For Each activity you added in step 19, and then click **OK**.
Be sure to select the appropriate property for the type of resource actual you are creating. For example, if you are creating an equipment actual, select the EquipmentActuals property.
 - c. Insert a Write activity inside the current For Each activity, and then add two writes.
 - d. Bind the first write to the workflow output parameter you configured in step 2d, bind the value of the write to the **DisplayName** property of the current For Each activity, and then click **OK**.
 - e. Bind the second write to the DisplayName property of the local variable you created in step 15b, bind the value of the write to the **DisplayName** property of the current For Each activity, and then click **OK**.
 - f. Insert an If/Else activity after the Write activity.
 - g. For the left branch of the If/Else activity, click **Edit** on the activity configuration panel to enter a unique name for the condition, and then click **Edit Condition** to enter an expression that represents the result the workflow should evaluate to; for example, `{Query Segment Responses}.{Equipment Display Name} == "01 - Motherboard Attach"`.
 - h. Insert a Write activity within the left branch of the If/Else activity after the condition, click **Edit** on the activity configuration activity panel to enter a unique name for the write activity, bind the write to the output parameter you created in step 2e, and then from the **Value** drop-down list, select `True`.
21. Click **Save**.

22. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.
23. You can view the resource actuals by navigating to the work request you created with this workflow and clicking the **Response Information** tab in the **S95 Model Editor**. For more information, see *Query segment responses*.

Add resource actual properties

Before you begin

After you create a workflow, you can add and configure actuals using call methods for service providers. You can create expressions or bind properties for many of these properties, depending on your actual requirements.



Note:

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with



Note:

A resource must already have been added in order to select it from this list.

3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.

The **Configure Method** dialog box appears, displaying a list of the available call method service providers.

7. Double-click the service provider you want to use, and then select the call method.
8. Click **Next**.
9. In the **Item specific configuration** section, configure the method.



Note:

Item configuration depends on the selected source and item. Some items do not require configuration.

10. At run time, if you want to return all data from the selected item, select the **Return all data** option.
11. At run time, if you want to return a subset of data from the selected item at run time, select the **Return only a subset of data** option, and then from the data tree, select the data you want returned.



Note:

If the item returns only a single piece of data, the **Return all data** option is selected, and you cannot retrieve a subset of data.

12. Click **Finish**.
13. If required, on the activity configuration panel, modify the inputs and/or create an expression or perform a bind for the selected call method.
14. In the IProductionRuntime methods, select **AddResourceActualProperties**.
15. Click **Finish**.
16. In the Actual ID field, enter a literal value, or click the drop-down arrow to perform a bind.



Note:

The data type of this field must be GUID.

17. In the Property Names field, perform one of the following:
 - a. Click the ellipsis button.
 - b. In the String Collection Editor dialog box, enter a series of strings for the names of the resource properties to add, and then click **OK**.

or

 - Click the drop-down arrow to perform a bind, and then click **OK**.

**Note:**

The data type of this field must be String[].

18. In the Resource Type field, select a resource type from the drop-down menu.
19. Click **Save**.

Write to resource actual properties

This procedure describes how to write resource actual properties to a specific resource.

Before you begin

You can create expressions or bind properties for many of these properties, depending on your actual requirements.

**Note:**

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click the service provider you want to use, and then select the call method.

8. Click **Next**.
9. In the **Item specific configuration** section, configure the method.

**Note:**

Item configuration depends on the selected source and item. Some items do not require configuration.

10. At run time, if you want to return all data from the selected item, select the **Return all data** option.
11. At run time, if you want to return a subset of data from the selected item at run time, select the **Return only a subset of data** option, and then from the data tree, select the data you want returned.
12. Click **Finish**.
13. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
14. Double-click **IProductionRuntime** to display the list of available call methods.
15. Click **WriteResourceActualProperties**, and then click **Next**.
16. Click **Browse resources for properties**.
The **Universal Browser** appears.
17. Select one of the resource models, select a resource, and then click **OK**.
18. Select the properties to write to, and then click **Finish**.
19. In the **Actual ID** field, enter a literal value, or click the drop-down arrow to perform a bind or create an expression.

**Note:**

The data type of this field must be GUID.

20. In the field for each selected resource property, enter a literal value, or click the drop-down arrow to perform a bind or create an expression.
21. From the **Resource Type** drop-down list, select a resource type to write to.
22. Click **Save**.
23. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

Move work responses to a new work request

You can move a work response from one work request to another work request at any time, regardless of the status of the originating work request.

Procedure

1. Create a new work request or ensure that you have a work request that you can move your work responses to.
2. In the navigator, click **Workflow > Workflows**.
3. In the resource list, select the workflow that you want to work with. If no workflow exists, create a new workflow.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
7. On the activity configuration panel, click **Configure Method**.

The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 8. Double-click **IProductionRuntime** to display the list of available call methods.
 9. Click **RepurposeWorkResponse**, and then click **Finish**.

The input parameters appear in the activity configuration panel.
 10. Configure the call method as follows:
 - a. In the **Destination Work Request** field, click the ellipsis button, and then select the work request whose work responses you want to move.
 - b. In the **Origin Work Request** field, click the ellipsis button, and then select the work request you want to move the work responses to.
 11. Click **Save**.
 12. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

Replace a segment requirement

You can create expressions or bind properties for many of these properties, depending on your workflow requirements.

About this task



Note:

When you replace a segment requirement, the entire hierarchy of the existing segment requirement is deleted and a new segment requirement with its associated hierarchy, if any, is created.

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 7. Double-click **IProductionRuntime** to display the list of available call methods.
 8. Click **ReplaceSegmentRequirement**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
 9. **Optional:** In the **Name** field, enter a name for the segment requirement, or bind a property or parameter to the method.

If unspecified, the new segment requirement will have the same name as the segment it replaces.

10. **Optional:** In the **Order** field, enter an execution order for the segment requirement, or bind a property or parameter to the method.

If unspecified, the new segment requirement will have the same order as the segment it replaces.

11. In the **Requirement ID** field, enter the ID of the segment requirement that will be deleted.
A new segment requirement will replace it at the same place in the hierarchy.

12. In the **Work Definition Segment** field, click the ellipsis button, and then select a work definition segment.

The segment requirement uses this definition to define properties and hierarchy.

13. Click **Save**.

14. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

Remove a resource actual from a segment response

This procedure describes all of the steps required to remove resource actuals from a specified segment response.

Before you begin

Depending on the type of actual you are creating, before you proceed with this procedure, make sure you have created the appropriate equipment, material, or personnel resource you are configuring the actual(s) for.

About this task

You can create expressions or bind properties for many of these properties, depending on your workflow requirements.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. Click **Add**, and then add and configure a workflow as follows:
 - a. On the workflow configuration panel, in the **Personnel** field, click the ellipsis button, and then select the personnel resource you want to assign to this workflow.
 - b. Click **Edit** next to **Parameters**.
The **Edit Parameters** dialog box appears.
 - c. Click **Add**, and then configure an input parameter with a **Type** of *String*, a **Direction** of *In*, and a unique name to bind to the **S95 Id** of the work request you will create in step 5b.

- d. Click **Add**, and then configure an output parameter to represent the resource actual's display name, with a **Type** of `String`, a **Direction** of `Out`, and a unique name to bind to the Write activity you will add in step 19d.
 - e. Click **Add**, and then configure an output parameter to represent the results of the workflow, with a **Type** of `Boolean`, a **Direction** of `Out`, and a unique name to bind to the Write activity you will add in step 19h.
 - f. Click **OK**.
 - g. Add a local subprocess to the workflow, click **Edit** on the subprocess configuration panel, enter a unique name to represent the resource actual configuration, and then click **OK**.
 - h. Add another local subprocess after the first, click **Edit** on the subprocess configuration panel, enter a unique name to represent the query you will configure in step 17, and then click **OK**.
3. Double-click the resource actual subprocess, and then insert a Call Method activity.
 4. On the activity configuration panel, click **Configure Method**.

The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 5. In the **IProductionRuntime** methods, click **CreateWorkRequest**, click **Finish**, and then configure the input parameters.
 - a. **Optional:** In the **Description** field, enter information to uniquely describe the work request.
 - b. In the **S95 Id** field, bind to the input parameter you configured for the workflow in step 2c.
 - c. In the **Work Definition** field, click the ellipsis button, and then select the work definition to associate with the work request.
 6. Insert another Call Method activity, and then click **Configure Method**.
 7. In the **IProductionRuntime** methods, click **CreateWorkResponse**, and then click **Next**.
 - a. In the **Item specific configuration** area, select a work type from the drop-down list, and then click **Finish**.
 - b. On the activity configuration panel, click **Edit**, enter a unique name to represent the call method, and then click **OK**.
 - c. **Optional:** In the **Description** field, enter information to uniquely describe the work response.
 - d. In the **Location** field, click the ellipsis button, and then select the equipment location to associate with the work response.

- e. In the **S95 Id** field, enter a name for the work response, or click the arrow and create an expression; for example, "`<workresponsename>`" + `ToString(CurrentDateTime())`, which will create a unique name for every work response that is created when this workflow is run. The value of "`<workresponsename>`" is the name you choose to give to the work response(s).
 - f. In the **Work Request** field, click the arrow, and then bind the work response to the work request you created in step 5.
 8. Insert another Call Method activity, and then click **Configure Method**.
 9. In the **IProductionRuntime** methods, select **CreateSegmentResponse**, and then click **Finish**.
 - a. On the activity configuration panel, click **Edit**, enter a unique name to represent the call method, and then click **OK**.
 - b. **Optional:** In the **Description** field, enter information to uniquely describe the segment response.
 - c. **Optional:** In the **End Time** field, click the arrow, and then select the terminating date and time for the segment response.
 - d. In the **Location** field, click the ellipsis button, and then select the equipment location to associate with the segment response.
 - e. In the **Parent Id** field, click the arrow, and then bind the segment response to the output parameter of the work response you created in step 6.
 - f. In the **S95 Id** field, enter a name for the segment response, or click the arrow and create an expression; for example, "`<segmentresponsename>`" + `ToString(CurrentDateTime())`, which will create a unique name for every segment response that is created when this workflow is run. The value of "`<workresponsename>`" is the name you choose to give to the segment response(s).
 - g. **Optional:** In the **Start Time** field, click the arrow, and then select the starting date and time for the segment response.
 - h. In the **Work Process Segment** field, click the ellipsis button, and then select the work process segment to associate with the segment response.
 10. Insert another Call Method activity, and then click **Configure Method**.
 11. In the **IProductionRuntime** methods, select one of the following call methods, and then click **Finish**:

- **CreateEquipmentActual**
- **CreateMaterialActual**
- **CreatePersonnelActual**

12. Configure the call method as follows:

- a. On the activity configuration panel, click **Edit**, enter a unique name to represent the call method, and then click **OK**.
- b. **Optional:** In the **Description** field, enter information to uniquely describe the resource actual you want to create.
- c. Depending on the type of resource actual you are creating, in the **Equipment**, **Material**, or **Personnel** field, click the ellipsis button, select the appropriate resource for the actual, and then click **OK**.
- d. **Optional:** If you are creating a material resource actual, then, in the **Lot or Sublot ID** field, enter a literal value, or click the drop-down arrow to perform a bind or create an expression.



Note:

The data type of this field must be GUID.

- e. In the **Quantity** field, enter a literal value, or click the drop-down arrow to perform a bind or create an expression.
- f. **Optional:** In the **Requirement ID** field, enter the resource requirement you want to associate with the actual, or click the drop-down arrow to perform a bind or create an expression.



Note:

The data type of this field must be GUID.

- g. In the **S95 ID** field, enter the unique name of the resource actual as you want it to appear on the segment response.
- h. In the **Segment Response ID** field, click the drop-down arrow and bind the value to the return value of the output parameter of the segment response you created in step 9.
- i. In the **Unit of Measure** field, enter the unit of measure you want to associate with the resource actual.

13. Click **Save**.

14. Return to the workflow level, double-click the subprocess you added in step 2h, and then configure the subprocess as follows:
 - a. On the subprocess configuration panel, click **Edit** next to **Local Variables**.
The **Edit Local Variables** dialog box appears.
 - b. Click **Add**, and then configure a variable with a **Type** of `String`, an **Initial Value** of `**`, and a unique name to bind to the Write activity you will create in step 19e.
 - c. Click **OK**.
15. Insert a Call Method activity.
16. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
17. In the **IProductionRuntime** methods, click **QuerySegmentResponses**, click **Finish**, and then configure the input parameters.
 - a. On the activity configuration panel, click **Edit**, enter a unique name to represent the call method, and then click **OK**.
 - b. From the **Detail Level** drop-down list, select `FullDetails`.
18. Insert a For Each activity after the QuerySegmentResponses Call Method activity, and then configure it as follows:
 - a. On the activity configuration panel, click **Edit**, enter a unique name to represent the activity, and then click **OK**.
 - b. In the **Collection** field, click the arrow, bind to the return value of the call method you added in step 9, and then click **OK**.
19. Insert another For Each activity after the first For Each activity, and then configure it as follows:
 - a. On the activity configuration panel, click **Edit**, enter a unique name to represent the activity, and then click **OK**.
 - b. In the **Collection** field, click the arrow, bind to the resource actual property of the For Each activity you added in step 18, and then click **OK**.
Be sure to select the appropriate property for the type of resource actual you are creating. For example, if you are creating an equipment actual, select the EquipmentActuals property.
 - c. Insert a Write activity inside the current For Each activity, and then add two writes.
 - d. Bind the first write to the workflow output parameter you configured in step 2d, bind the value of the write to the **DisplayName** property of the current For Each activity, and then click **OK**.

- e. Bind the second write to the `DisplayName` property of the local variable you created in step 14b, bind the value of the write to the **DisplayName** property of the current For Each activity, and then click **OK**.
 - f. Insert an If/Else activity after the Write activity.
 - g. For the left branch of the If/Else activity, click **Edit** on the activity configuration panel to enter a unique name for the condition, and then click **Edit Condition** to enter an expression that represents the result the workflow should evaluate to; for example, `{Query Segment Responses}.{Equipment Display Name} == "01 - Motherboard Attach"`.
 - h. Insert a Write activity within the left branch of the If/Else activity after the condition, click **Edit** on the activity configuration panel to enter a unique name for the write activity, bind the write to the output parameter you created in step 2e, and then from the **Value** drop-down list, select `True`.
20. In the **IProductionRuntime** methods, select one of the following call methods, and then click **Finish**:
- **DeleteEquipmentActual**
 - **DeleteMaterialActual**
 - **DeletePersonnelActual**
21. Configure the call method as follows:
- a. In the **Actual ID** field, click the drop-down arrow, bind the value to the **Id** parameter of the subprocess you created in step 2g, and then click **OK**.
 - b. In the **Segment Id** field, click the drop-down arrow, bind the value to the `Id` property of the Query Segment Response subprocess you created in step 2h.
22. Click **Save**.
23. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

Production Runtime Call Methods

The `IProductionRuntime` service interface call methods allow you to create, modify, query, and delete work requests, work responses, segment responses, actuals, and other items related to production oversight.

On the activity configuration panel, you can use a *bind* or *expression* to define your input and output parameters.

Subtopics in this section describe each service method for this service interface.

AddRequestMaterialBillItem (IProductionRuntime)

Adds an item as a request material bill item to the request bill of materials (BOM) for the selected work request.

Inputs	Data Type	Description
materialRequirementName	String	Specifies the name of the material requirement to add to the request bill of materials (BOM).
requestMaterialBillItemName	String	Specifies the name for the new request bill of materials (BOM) item.
requestMaterialBillName	String	Specifies the name of the request bill of materials (BOM) to add the material specification to.
segmentId	GUID	Specifies the ID of the segment requirement the material requirement is associated with.
workRequestAddress	DirectoryResource	Specifies the work request to add the request bill of materials (BOM) item to.

Outputs	Description
OperationsMaterialBillItemAttributes	Returns attributes of the newly added request bill of materials (BOM) item.

AddResourceActualProperties (IProductionRuntime)

Adds named properties of the resource as resource actual properties.

Inputs	Data Type	Description
Actual ID	GUID	Specifies the value for the resource actual to add the proper-

Inputs	Data Type	Description
		<p>ties to. This is the ID that is returned when you create a segment response.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>
Property Names	String[]	<p>Specifies the names of the properties' input parameters to add.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>
Resource Type	Dependent on each parameter's data type	<p>Specifies the resource type of the resource actual being written to. There are three options:</p> <ul style="list-style-type: none"> • Material • Equipment • Personnel

AddResourceRequirementProperties (IProductionRuntime)

Adds named properties of the resource as resource requirement properties.

Inputs	Data Type	Description
Property Names	String[]	<p>Specifies the names of the properties' input parameters to add.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Uni-</p>

Inputs	Data Type	Description
		versal Browser to select the resource you want to use.
Requirement Id	GUID	Specifies the value for the resource requirement to add the properties to. This is the ID that is returned when you create a segment response. In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.
Resource Type	Dependent on each parameter's data type	Specifies the resource type of the resource actual being written to. There are three options: <ul style="list-style-type: none"> • Material • Equipment • Personnel

AddWorkDataClassToResponse (IProductionRuntime)

Adds work data from a work data class to a segment response.



Note:

If you add a work data class that the process segment does not have, when you try to write a value to it, the write fails and no values are recorded. The data class must be an optional data class in the process segment definition.

Inputs	Data Type	Description
Segment Response ID	String	Specifies the segment response you want to add a work data class to.

Inputs	Data Type	Description
		In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.
Work Data Class	String	Specifies the work data class to add to the segment response. In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.

CreateEquipmentActual (IProductionRuntime)

Creates an equipment actual on a specified segment response, without the ability to set initial start and end time values for the actual.

If you want to set initial start and end time values for the actual, use the *CreateEquipmentActualEx* call method instead.



Note:

The S95 ID must be unique in the system or the call method fails.

Inputs	Data Type	Description
Description	String	Specifies the name of the equipment actual.
Equipment	DirectoryResource	Specifies the equipment or equipment class associated with the equipment actual. In the Inputs section of the activity configuration panel, click the ellipsis button to open the Uni-

Inputs	Data Type	Description
		versal Browser to select the resource you want to use.
Quantity	Double	Specifies the quantity of the equipment actual.
Requirement ID (optional)	GUID	Specifies the resource requirement you want to associate with the equipment actual.
S95 ID	String	Specifies the unique name of the equipment actual on the specified segment response.
Segment Response ID	GUID	Specifies the segment response you want to associate with the equipment actual.
Unit of Measure	String	Specifies the unit of measure you want to associate with the equipment actual.

Outputs	Description
GUID	Returns the new equipment actual.

Related reference

[CreateEquipmentActualEx \(IProductionRuntime\) \(on page 1938\)](#)

CreateEquipmentActualEx (IProductionRuntime)

Creates an equipment actual on a specified segment response, with the ability to set initial start and end time values for the actual.



Note:

The S95 ID must be unique in the system or the call method fails.

Inputs	Data Type	Description
Description	String	Specifies the name of the equipment actual.
End Time	DateTime	<p>Specifies the initial end date and time, in UTC, for the actual.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>
Equipment	DirectoryResource	<p>Specifies the equipment or equipment class associated with the equipment actual.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>
Quantity	Double	Specifies the quantity of the equipment actual.
Requirement ID (optional)	GUID	Specifies the resource requirement you want to associate with the equipment actual.
S95 ID	String	Specifies the unique name of the equipment actual on the specified segment response.
Segment Response ID	GUID	Specifies the segment response you want to associate with the equipment actual.
Start Time	DateTime	Specifies the initial start date and time, in UTC, for the actual.

Inputs	Data Type	Description
		In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.
Unit of Measure	String	Specifies the unit of measure you want to associate with the equipment actual.

Outputs	Description
GUID	Returns the new equipment actual.

CreateMaterialActual (IProductionRuntime)

Creates a material actual on a specified segment response, without the ability to set initial start and end time values for the actual.

If you want to set initial start and end time values for the actual, use the *CreateMaterialActualEx* call method instead.



Note:

- The S95 ID must be unique in the system or the call method fails.
- The material actual can be associated with only one of the following: a Lot ID, Sublot ID, or a Material resource.

Inputs	Data Type	Description
Description	String	Specifies the description of the material actual.
Lot or Sublot ID	GUID	Specifies the lot or sublot ID of the material lot or material sublot

Inputs	Data Type	Description
		associated with the material actual.
Material	DirectoryResource	Specifies the name of the material definition or material class associated with the material actual. In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.
Quantity	Double	Specifies the quantity of the material actual.
Requirement ID	GUID	Specifies the resource requirement you want to associate with the material actual.
S95Id	String	Specifies the unique name of the material actual on the specified segment response.
Segment Response ID	GUID	Specifies the segment response you want to associate with the material actual.
Unit of Measure	String	Specifies the unit of measure for the quantity you want to associate with the material actual.

Outputs	Description
GUID	Returns the new material actual.

Related reference

[CreateMaterialActualEx \(IProductionRuntime\) \(on page 1942\)](#)

CreateMaterialActualEx (IProductionRuntime)

Creates a material actual on a specified segment response, with the ability to set initial start and end time values for the actual.



Note:

- The S95 ID must be unique in the system or the call method fails.
- The material actual can be associated with only one of the following: a Lot ID, Sublot ID, or a Material resource.

Inputs	Data Type	Description
Description	String	Specifies the description of the material actual.
End Time	DateTime	Specifies the initial end date and time, in UTC, for the actual. In the Inputs section of the activity configuration panel, click the drop-down arrow to set the date and time.
Lot or Sublot ID	GUID	Specifies the lot or subplot ID of the material lot or material subplot associated with the material actual.
Material	DirectoryResource	Specifies the name of the material definition or material class associated with the material actual. In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.

Inputs	Data Type	Description
Quantity	Double	Specifies the quantity of the material actual.
Requirement ID	GUID	Specifies the resource requirement you want to associate with the material actual.
S95 ID	String	Specifies the unique name of the personnel actual on the specified segment response.
Segment Response ID	GUID	Specifies the segment response you want to associate with the material actual.
Start Time	DateTime	Specifies the initial start date and time, in UTC, for the actual. In the Inputs section of the activity configuration panel, click the drop-down arrow to set the date and time.
Unit of Measure	String	Specifies the unit of measure for the quantity you want to associate with the material actual.

Outputs	Description
GUID	Returns the new material actual.

CreateMaterialRequirement (IProductionRuntime)

Creates a material requirement for a segment requirement.

Inputs	Data Type	Description
Description	String	Specifies a description for the material requirement.

Inputs	Data Type	Description
Material	DirectoryResource	<p>Specifies the name of the material definition or material class associated with the material requirement.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>
Material Use	MaterialUses	<p>Specifies how the material requirement will be used. Valid values are:</p> <ul style="list-style-type: none"> • Consumed • Produced • Consumable
Name	DirectoryResource	<p>Specifies the name of the material definition or material class associated with the material requirement.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>
Quantity	Double	Specifies the quantity of the material resource requirement used.
Segment Id	GUID	Specifies the ID of the segment you want to associate with the material requirement.
Unit of Measure	String	Specifies the unit of measure for the quantity you want to asso-

Inputs	Data Type	Description
		ciate with the material requirement.

Outputs	Description
GUID	Returns the ID of the new material requirement.

CreatePersonnelActual (IProductionRuntime)

Creates a personnel actual on a specified segment response, without the ability to set initial start and end time values for the actual.

If you want to set initial start and end time values for the actual, use the *CreatePersonnelActualEx* call method instead.



Note:

The S95 ID must be unique in the system or the call method fails.

Inputs	Data Type	Description
Description	DirectoryResource	Specifies the description of the personnel actual.
Personnel	String	Specifies the name of the person or personnel class associated with the personnel actual. In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.
Quantity	Double	Specifies the quantity of the personnel resource actual used.

Inputs	Data Type	Description
Requirement ID	GUID	Specifies the resource requirement you want to associate with the personnel actual.
S95 ID	String	Specifies the unique name of the personnel actual on the specified segment response.
Segment Response ID	GUID	Specifies the segment response you want to associate with the personnel actual.
Unit of Measure	String	Specifies the unit of measure for the quantity you want to associate with the personnel actual.

Outputs	Description
GUID	Returns the new personnel actual.

Related reference

[CreatePersonnelActualEx \(IProductionRuntime\) \(on page 1946\)](#)

CreatePersonnelActualEx (IProductionRuntime)

Creates a personnel actual on a specified segment response, with the ability to set initial start and end time values for the actual.



Note:

The S95 ID must be unique in the system or the call method fails.

Inputs	Data Type	Description
Description	DirectoryResource	Specifies the description of the personnel actual.
End Time	DateTime	Specifies the initial end date and time, in UTC, for the actual.

Inputs	Data Type	Description
		In the Inputs section of the activity configuration panel, click the drop-down arrow to set the date and time.
Personnel	String	<p>Specifies the name of the person or personnel class associated with the personnel actual.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>
Quantity	Double	Specifies the quantity of the personnel resource actual used.
Requirement ID	GUID	Specifies the resource requirement you want to associate with the personnel actual.
S95 ID	String	Specifies the unique name of the personnel actual on the specified segment response.
Segment Response ID	GUID	Specifies the segment response you want to associate with the personnel actual.
Start Time	DateTime	<p>Specifies the initial start date and time, in UTC, for the actual.</p> <p>In the Inputs section of the activity configuration panel, click the drop-down arrow to set the date and time.</p>
Unit of Measure	String	Specifies the unit of measure for the quantity you want to associate with the personnel actual.

Outputs	Description	
GUID		Returns the new personnel actual.

CreateRequestMaterialBill (IProductionRuntime)

Creates a request bill of materials (BOM) for the selected work request.

Inputs	Data Type	Description
Description	String	Specifies the description for the new request bill of materials (BOM).
Name	String	Specifies the name for the new request bill of materials (BOM).
workRequestAddress	DirectoryResource	Specifies the work request to create the request bill of materials (BOM) for.

Outputs	Description	
RequestMaterialBillAttributes		Returns the attributes of the request bill of materials (BOM) item.

CreateSegmentRequirement (IProductionRuntime)

Creates a new segment requirement that is based upon a supplied work definition segment.

Inputs	Data Type	Description
Name	String	Specifies the name of the segment requirement.
Order	Int	Specifies the execution order of the segment requirement.

Inputs	Data Type	Description
Parent Requirement ID	GUID	(Optional) Specifies the ID of the parent segment requirement the new segment requirement is associated with.
Work Definition Segment	DirectoryResource	Specifies the work definition segment to associate with the segment requirement. In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.
Work Request	DirectoryResource	Specifies the work request to associate with the segment requirement. In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.

Outputs	Description
GUID	Returns the new segment requirement.

Related information

[Create a segment requirement \(on page 1911\)](#)

CreateSegmentResponse (IProductionRuntime)

Creates a segment response to associate with a work request or parent segment response, and work process segment.

Inputs	Data Type	Description
Description	String	Specifies the description of the segment response.
End Time	DateTime	<p>Specifies the end date and time for the segment response.</p> <p>In the Inputs section of the activity configuration panel, click the drop-down arrow to set the date and time.</p>
Location	DirectoryResource	<p>Specifies the equipment location to associate with the segment response.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>
Parent ID	GUID	<p>Specifies the ID of the work response associated with the segment response you are creating. Bind this parameter to the ReturnValue property of the work response.</p> <p>- Or -</p> <p>Specifies the ID of the segment response that you want to specify as the parent segment. Bind this parameter to a ReturnValue property of a segment response to create a child of that segment response.</p>

Inputs	Data Type	Description
		 Note: A child segment response will automatically inherit its parent's work response.
S95 ID	String	Specifies the name of the segment response.
Start Time	DateTime	Specifies the start date and time for the segment response. In the Inputs section of the activity configuration panel, click the drop-down arrow to set the date and time.
Work Process Segment	DirectoryResource	Specifies the work process segment that defines the work data classes that are available to be written to by this segment response. In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.

Outputs	Description
GUID	Returns the new segment response.

CreateSegmentResponseForRequirement (IProductionRuntime)

Creates a segment response for a specified work response or parent segment response and segment requirement.

Inputs	Data Type	Description
Description	String	Specifies a description for the segment response.
End Time	DateTime	<p>Specifies the end date and time for the segment response.</p> <p>In the Inputs section of the activity configuration panel, click the drop-down arrow to set the date and time.</p>
Location	DirectoryResource	<p>Specifies the equipment location to associate with the segment response.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>
Parent ID	GUID	<p>Specifies the ID of the work response associated with the segment response you are creating. Bind this parameter to the ReturnValue property of the work response.</p> <p>- Or -</p> <p>Specifies the ID of the segment response that you want to specify as the parent segment. Bind this parameter to a ReturnValue property of a segment response to create a child of that segment response.</p>

Inputs	Data Type	Description
		 Note: A child segment response will automatically inherit its parent's work response.
Requirement ID	GUID	Specifies the name of the segment requirement.
S95 ID	String	Specifies the ID of the segment response.
Start Time	DateTime	Specifies the start date and time for the segment response. In the Inputs section of the activity configuration panel, click the drop-down arrow to set the date and time.

Outputs	Description
GUID	Returns the ID of the new segment response.

CreateWorkRequest (IProductionRuntime)

Creates a new work request based on a specified work definition.



Note:

The S95 ID must be unique in the system or the call method fails.

Inputs	Data Type	Description
Description	String	Specifies the description of the work request.

Inputs	Data Type	Description
S95 ID	String	Specifies the name of the work request.
Work Definition	DirectoryResource	Specifies the work definition you want to base the work request on. In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.

Outputs	Description
DirectoryResource	Returns the new work request.

CreateWorkResponse (IProductionRuntime)

Creates a work response based on a specified work definition.



Note:

Multiple process segment responses can be associated with a work response.

Inputs	Data Type	Description
Description	String	Specifies the description of the work response.
Location	DirectoryResource	Specifies the equipment location to associate with the work response. In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.

Inputs	Data Type	Description
S95 ID	String	Specifies the name of the work response.
Work Request	DirectoryResource	Specifies the work request to associate with the work response. This can be empty if there are no work requests to associate with this response.
Work Type	String	<p>Specifies the work type of the work request you are associating with the work response.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • Production • Maintenance • Quality • Inventory • Mixed • Unspecified <p>On the Configure Method dialog box, click Next to select a work type. You cannot select or enter a work type on the activity configuration panel.</p>

Outputs	Description	
GUID		Returns the new work response.

DeleteEquipmentActual (IProductionRuntime)

Deletes an equipment actual associated with a specific segment response.

If you require resource actuals to be created but do not need to keep them on a segment response, you can delete the actuals immediately after creating them.

Property	Value	Description
Actual ID	GUID	Specifies the unique ID of the resource actual you want to delete. Bind this parameter to the Id property of the subprocess used to create the resource actual.
Segment ID	GUID	Specifies the unique ID of the segment response associated with the resource actual you want to delete. Bind this parameter to the Id property of the subprocess used to query the segment responses.

DeleteMaterialActual (IProductionRuntime)

Deletes a material actual associated with a specific segment response.

If you require resource actuals to be created but do not need to keep them on a segment response, you can delete the actuals immediately after creating them.

Property	Value	Description
Actual ID	GUID	Specifies the unique ID of the resource actual you want to delete. Bind this parameter to the Id property of the subprocess used to create the resource actual.
Segment ID	GUID	Specifies the unique ID of the segment response associated with the resource actual you want to delete. Bind this parameter to the Id property of the subprocess used to query the segment responses.

DeletePersonnelActual (IProductionRuntime)

Deletes a personnel actual associated with a specific segment response.

If you require resource actuals to be created but do not need to keep them on a segment response, you can delete the actuals immediately after creating them.

Property	Value	Description
Actual ID	GUID	Specifies the unique ID of the resource actual you want to delete. Bind this parameter to the Id property of the subprocess used to create the resource actual.
Segment Id	GUID	Specifies the unique ID of the segment response associated with the resource actual you want to delete. Bind this parameter to the Id property of the subprocess used to query the segment responses.

DeleteResourceRequirement (IProductionRuntime)

Deletes a resource requirement.

Inputs	Data Type	Description
Requirement ID	GUID	Specifies the ID of the resource requirement to delete.
Resource Type	ResourceTypes	Specifies the category of the selected specification. Valid values are: <ul style="list-style-type: none"> • Material • Equipment • Personnel

DeleteSegmentResponse (IProductionRuntime)

Deletes a specific segment response.

Property	Value	Description
Response ID	GUID	<p>Specifies the unique ID of the segment response you want to delete.</p> <p>Use this call method with the QuerySegmentResponse call method and an Index activity with an integer value. You can then bind this parameter to one of the Index activity's properties.</p>

DeleteWorkRequest (IProductionRuntime)

Deletes a specific work request.

Property	Value	Description
Work Request	DirectoryResource	Specifies the work request you want to delete.

GetMasterSegmentForWorkRequest (IProductionRuntime)

Gets the master segment for a work request.



Note:

The master segment for a work request is a segment requirement.

Inputs	Data Type	Description
Work Request	DirectoryResource	<p>Specifies the work request whose master segment you want to retrieve.</p> <p>In the Inputs section of the activity configuration panel, click the</p>

Inputs	Data Type	Description
		ellipsis button to open the Universal Browser to select the resource you want to use.

Outputs	Description
GUID	Returns the master segment (that is, the segment requirement) for the specified work request.

GetResourceForWorkRequest (IProductionRuntime)

Gets the resource specification that is linked to a work request.

Inputs	Data Type	Description
Work Request	DirectoryResource	Specifies the work request whose resource specification you want to retrieve. In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.

Outputs	Description
DirectoryResource	Returns the resource specifications (that is, the equipment, material, and/or personnel specifications) for the specified work request.

GetWorkRequestLinksForResource (IProductionRuntime)

Gets all work requests linked to a resource specification.

Inputs	Data Type	Description
Linked Resource	DirectoryResource	Specifies the resource specification whose work request(s) you want to retrieve.

Outputs	Description
DirectoryResource[]	Returns all of the work requests that the specified resource specification is linked to.

ReadRequestMaterialBill (IProductionRuntime)

Reads the attributes of a request bill of materials (BOM) associated with the selected work request.

Inputs	Data Type	Description
requestMaterialBillName	String	Specifies the name of the request bill of materials (BOM) to read.
workRequestAddress	DirectoryResource	Specifies the work request associated with the request bill of materials to be read.

Outputs	Description
RequestMaterialBillAttributes	Returns the attributes of the selected request bill of materials (BOM).

ReadSegmentRequirement (IProductionRuntime)

Reads the attributes of a specific segment requirement.

Inputs	Data Type	Description
Detail Level	SegmentRequirementDetailLevel	Specifies the detail level of the output. Valid values are: <ul style="list-style-type: none"> • SegmentsOnly • IncludeParameters • IncludeEquipmentRequirements • IncludeMaterialRequirements • IncludePersonnelRequirements • ChildrenOnly • FullDetails
Reurement ID	GUID	Specifies the ID of the segment requirement whose attributes you want to read.

Outputs	Description
SegmentRequirementAttributes	Returns the segment requirement attributes.

ReadSegmentRequirements (IProductionRuntime)

Reads the attributes of segment requirements for a specific work request.

Inputs	Data Type	Description
Detail Level	SegmentRequirementsDetails-Level	Specifies the detail level of the output. Valid values are: <ul style="list-style-type: none"> • SegmentsOnly • IncludeParameters • IncludeEquipmentRequirements • IncludeMaterialRequirements

Inputs	Data Type	Description
		<ul style="list-style-type: none"> • IncludePersonnelRequirements • ChildrenOnly • FullDetails
Work Request	DirectoryResource	<p>Specifies the work request with the segment requirement attributes you want to read.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>

Outputs	Description	
SegmentRequirementCollection		<p>Returns the segment requirement attributes.</p> <p>Use a <i>For Each Activity</i> to loop through all resources or use an <i>Index Activity</i> to find a named segment.</p> <div data-bbox="1024 1304 1421 1522" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: Work parameters are the properties that can be indexed. </div>



Important:

The returned collection of data is based on user-defined values and is dependent on your published data. Each segment requirement can contain work parameters, as well as material, equipment, and personnel requirements. For information on the attributes that can be read from, see *Work Requests*.

ReadSegmentResponse (IProductionRuntime)

Reads the attributes of the specified segment response.

Inputs	Data Type	Description
Detail Level	SegmentResponseDetailLevel	Specifies the detail level of the output. Valid values are: <ul style="list-style-type: none"> • AttributesOnly • IncludeWorkData • IncludeEquipmentActuals • IncludeMaterialActuals • IncludePersonnelActuals • ChildrenOnly • FullDetails
Response ID	GUID	Specifies the ID of the segment response whose attributes you want to read.

Outputs	Description
SegmentResponseAttributes	Returns the segment response attributes.

ReadWorkData (IProductionRuntime)

Reads the attributes of a work data class that is associated with a specific segment response.



Note:

In the Configure Method dialog box, click Next to select a work data class and its properties that you want to read. You can select all properties or a subset. If you do not select a work data class or properties, all work data classes associated with the segment response are returned.

Inputs	Data Type	Description
Response ID	GUID	Specifies the GUID value for the segment response. This is the ID

Inputs	Data Type	Description
		that is returned when you create a segment response.

Outputs	Description
WorkDataCollection	<p>A collection of properties from a work data class.</p> <p>Use an <i>Index Activity</i> based on name, or use a <i>For Each Activity</i> to loop through the properties</p>



Important:

The returned collection of data is based on user-defined values and is dependent on your published data. For information on the attributes that can be read from, see *Work Data Classes*.

ReadWorkRequestAttributes (IProductionRuntime)

Reads the attributes of a specific work request.

Inputs	Data Type	Description
Work Request	DirectoryResource	<p>Specifies the work request with the attributes you want to read.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>

Outputs	Description
WorkRequestAttributes	Attribute values of the work request.

**Important:**

The returned data is based on user-defined values and is dependent on your published data. For information on the attributes that can be read from, see *Work Requests*.

ReadWorkRequestMaterialBills (IProductionRuntime)

Inputs	Data Type	Description
workRequestAddress	DirectoryResource	Specifies the work request whose request bills of material are read.

Outputs	Description
RequestMaterialBillAttributes[]	Returns a collection of request bills of material (BOM) attributes.

RemoveRequestMaterialBill (IProductionRuntime)

Removes the specified request bill of materials (BOM) from the selected work request.

Inputs	Data Type	Description
requestMaterialBillName	String	Specifies the name of the request bill of materials (BOM) to remove.
workRequestAddress	DirectoryResource	Specifies the work request to remove the request bill of materials (BOM) from.

RemoveRequestMaterialBillItem (IProductionRuntime)

Removes a material requirement from the specified request bill of materials (BOM) for the selected work request.

Inputs	Data Type	Description
requestMaterialBillItemName	String	Specifies the name of the Request bill of materials (BOM) to remove the item from.
requestMaterialBillName	String	Specifies the name of the Request bill of materials (BOM) to remove the item from.
workRequestAddress	DirectoryResource	Specifies the work request associated with the request bill of materials (BOM) the item is being removed from.

ReplaceSegmentRequirement (IProductionRuntime)

Deletes the entire hierarchy of the existing segment requirement and creates a new segment requirement with any associated hierarchy.

Inputs	Data Type	Description
Name	String	Specifies the name of the segment requirement. If a value is not specified, the new segment requirement will have the same name as the one it replaces.
Order	Int	(Optional) Specifies the execution order of the segment requirement. If a value is not specified, the new segment requirement will have the same order as the one it replaces.
Requirement ID	GUID	Specifies the ID of the segment requirement that will be deleted and then replaced by the cur-

Inputs	Data Type	Description
		rently configured one. The new segment requirement takes the same place in the hierarchy.
Work Definition Segment	DirectoryResource	Specifies the work definition segment to associate with the segment requirement. In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.

Outputs	Description
GUID	Returns the new segment requirement.

RepurposeWorkResponses (IProductionRuntime)

Moves the work responses from one work request to another work request, regardless of the status of the originating work request.



Note:

You can move all work responses for a work request; you cannot move a subset of work responses.

Inputs	Data Type	Description
Destination Work Request	DirectoryResource	Specifies the work request you want to move the work responses to.
Origin Work Request	DirectoryResource	Specifies the work request you want to move the work responses from.

Outputs	Description	
WorkResponseCollection		Returns the set of work responses that were moved from the original work request to the destination work request.

SetMaterialActualLotOrSublot (IProductionRuntime)

Sets a lot or subplot specified by a material actual.

SectionTitle

Attribute	Value	Description
Actual ID	GUID	Specifies the ID of the material actual to set.
Lot or Sublot ID	GUID	Specifies the ID of the material lot or subplot to set the material actual to.
Lot Type	LotType	Specifies whether the material actual is being set to a lot or subplot.

SetMaterialRequirementLotOrSublot (IProductionRuntime)

Sets a lot or subplot for the specified material requirement.

This call method is related to the SetResourceRequirement call method. While this call method is used to set either a material lot or subplot on a material requirement, the SetResourceRequirement call method is used to set a material definition on the requirement.



Important:

If you set a material definition on a requirement and then set a material lot (or subplot) on the same requirement, when you read the requirement attributes (ReadSegmentRequirements call method), the value that is returned is the value of the lot (or subplot). Therefore, whatever was last set on that material requirement, by either method, is what is returned when read.

Inputs	Data Type	Description
Lot or Sublot ID	GUID	Specifies the ID of the lot or subplot you want the specified material requirement to use.
Lot Type	String	Specifies whether you are setting a lot or subplot for the material requirement.
Requirement ID	GUID	Specifies the ID of the material requirement you are setting.

SetResourceActualEndTime (IProductionRuntime)

Updates the utilization end time for a previously created material, personnel, or equipment actual.

Inputs	Data Type	Description
Actual ID	GUID	Specifies the actual that utilization end time is being recorded for. In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.
Resource Type	String	Specifies the type of resource that the utilization end time is being recorded for. Valid options are: <ul style="list-style-type: none"> • Material • Equipment • Personnel
End Time	DateTime	Specifies the actual utilization end date and time, in UTC.

Inputs	Data Type	Description
		In the Inputs section of the activity configuration panel, click the drop-down arrow to set the date and time.

SetResourceActualStartTime (IProductionRuntime)

Updates the utilization start time for a previously created material, personnel, or equipment actual.

Inputs	Data Type	Description
Actual ID	GUID	<p>Specifies the actual that utilization start time is being recorded for.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>
Resource Type	Dependent on each parameter's data tuupe	<p>Specifies the type of resource that the utilization start time is being recorded for. Valid options are:</p> <ul style="list-style-type: none"> • Material • Equipment • Personnel
Start Time	DateTime	<p>Specifies the actual utilization start date and time, in UTC.</p> <p>In the Inputs section of the activity configuration panel, click the drop-down arrow to set the date and time.</p>

SetResourceActualUse (IProductionRuntime)

Property	Value	Description
Actual ID	GUID	Specifies the ID of the resource actual that you want to define a purpose for.
Resource Type	ResourceTypes	Specifies the type of resource actual to define a purpose for.
Use	String	Specifies the purpose for the defined resource actual. <div data-bbox="1024 674 1419 989" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: The value you enter in this field should be based on the resource actual uses defined by your business. </div>

SetResourceRequirement (IProductionRuntime)

Sets the resource specified by a resource requirement.

This call method is related to the SetMaterialRequirementLotOrSublot call method. While this call method is used to set a material definition on a material requirement, the SetMaterialRequirementLotOrSublot call method is used to set either a material lot or subplot on the requirement.



Important:

If you set a material definition on a requirement and then set a material lot (or subplot) on the same requirement, when you read the requirement attributes (ReadSegmentRequirements call method), the value that is returned is the value of the lot (or subplot). Therefore, whatever was last set on that material requirement, by either method, is what is returned when read.

Inputs	Data Type	Description
Requirement ID	GUID	Specifies the ID of a resource in the current process segment. This is available only by reading

Inputs	Data Type	Description
		segment requirements with resource requirements.
Resource	DirectoryResource	Specifies the resource to set. In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.
Resource Type	String	Specifies the type of resource to set. In the Inputs section of the activity configuration panel, click the drop-down arrow to select one of the following valid values: <ul style="list-style-type: none"> • Material • Equipment • Personnel

SetResourceRequirementQuantity (IProductionRuntime)

Sets the quantity specified by a resource requirement.

Inputs	Data Type	Description
Quantity	Int32	Specifies the quantity of the resource requirement to set.
Requirement ID	GUID	Specifies the ID of a resource in the current process segment. This is available only by reading segment requirements with resource requirements.
Resource Type	String	Specifies the type of resource to set.

Inputs	Data Type	Description
		<p>In the Inputs section of the activity configuration panel, click the drop-down arrow to select one of the following valid values:</p> <ul style="list-style-type: none"> • Material • Equipment • Personnel

SetResponseEndTime (IProductionRuntime)

Sets the end date and time for the work response or segment response.

Inputs	Data Type	Description
End Time	DateTime	<p>Specifies the end date and time of the work response or segment response.</p> <p>In the <i>Inputs</i> section of the activity configuration panel, click the drop-down arrow to set the date and time.</p>
Response ID	String	Specifies the ID of the work response or segment response you want to set the end date and time for.

SetResponseLocation (IProductionRuntime)

Sets the location of a work response or segment response.

Inputs	Data Type	Description
Response ID	GUID	Specifies the ID of the work response or segment response you want to set the location for.

Inputs	Data Type	Description
location	DirectoryResource	Specifies the equipment location of the response. Select null to clear the location to a response or reset it to the empty default.

SetResponseS95Id (IProductionRuntime)

Sets the S95 ID of a work response or segment response.

Inputs	Data Type	Description
responseld	GUID	Specifies the ID of the work response or segment response you want to set.
s95Id	String	Specifies the name to set on the response.

SetResponseStartTime (IProductionRuntime)

Sets the start date and time for the work response or segment response.

Inputs	Data Type	Description
Response ID	String	Specifies the ID of the work response or segment response you want to set the start date and time for.
Start Time	DateTime	Specifies the start date and time of the work response or segment response. In the Inputs section of the activity configuration panel, click the drop-down arrow to set the date and time.

SetSegmentRequirementEndTime (IProductionRuntime)

Sets the scheduled latest end time of the segment requirement.

Inputs	Data Type	Description
endTime	DateTime	Specifies the latest expected end date and time of the segment requirement in UTC.
Requirement ID	GUID	Specifies the ID of the segment requirement you are setting the end date and time for.



Note:

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

SetSegmentRequirementOrder (IProductionRuntime)

Sets the order of a specified segment requirement.

Attribute	Value	Description
Order	Int32	Specifies the order to set for the specified segment requirement.
segmentId	GUID	Specifies the ID of the segment you want to set the order for.

SetSegmentRequirementS95Id (IProductionRuntime)

Sets the S95Id of a specified segment requirement.

Attribute	Value	Description
S95Id	String	Specifies the S95 ID you want to apply to the segment requirement.

Attribute	Value	Description
segmentId	GUID	Specifies the ID of the segment you want to set the S95Id for.

SetSegmentRequirementStartTime (IProductionRuntime)

Sets the scheduled earliest start time and end time of the segment requirement.

Inputs	Data Type	Description
Requirement ID	GUID	Specifies the ID of the segment requirement you are setting the start date and time for.
startTime	DateTime	Specifies the earliest expected date and start time of the segment requirement, in UTC.



Note:

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

SetWorkRequestLocation (IProductionRuntime)

Sets the equipment location for a specific work request.

Inputs	Data Type	Description
Location	DirectoryResource	Specifies the equipment location for the work request. In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.
Work Request	DirectoryResource	Specifies the work request to set the location for. In the Inputs section of the activity configuration panel, click the

Inputs	Data Type	Description
		ellipsis button to open the Universal Browser to select the resource you want to use.

SetWorkRequestPriority (IProductionRuntime)

Sets the priority for a specified work request. Priorities are typically used for scheduling purposes.

Inputs	Data Type	Description
Priority	Int	Specifies the priority number of the work request. <div data-bbox="1024 793 1419 972" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;">  Note: The priority number can be a negative value. </div>
Work Request	DirectoryResource	Specifies the work request you are setting the priority for.

SetWorkRequestScheduledTimes (IProductionRuntime)

Sets the scheduled start and end date and time for a work request.

Inputs	Data Type	Description
End Time	DateTime	Specifies the scheduled end date and time to set for the work request. In the Inputs section of the activity configuration panel, click the drop-down arrow to set the date and time.
Start Time	DateTime	Specifies the scheduled start date and time to set for the work request.

Inputs	Data Type	Description
		In the Inputs section of the activity configuration panel, click the drop-down arrow to set the date and time.
Work Request	DirectoryResource	<p>Specifies the work request to set the start and end dates and times for.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>

SetWorkRequestStatus (IProductionRuntime)

Sets the status of a specific work request.

Inputs	Data Type	Description
Status	String	<p>Specifies the status to set for the work request.</p> <p>In the Configure Method dialog box, click Next to select the status you want to set the work request to. The value you select displays as an input parameter on the activity configuration panel. Valid values are:</p> <ul style="list-style-type: none"> • Draft • Unscheduled • Pending • Running • Paused • Completed

Inputs	Data Type	Description
		<ul style="list-style-type: none"> • Failed • Canceled • Unknown <p>You must select a status in the Configure Method dialog box. You cannot select or enter a status on the activity configuration panel.</p>
Work Request	DirectoryResource	<p>Specifies the work request you want to set the status for.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>

SetWorkResponseAssociatedWorkRequest (IProductionRuntime)

Sets the associated work request of a work response.

SectionTitle

Attribute	Value	Description
Work Request	DirectoryResource	Specifies the work request you want to associate with the work response.
Work Response	GUID	Specifies the ID of the work response to associated the work request with.

SetWorkResponseState (IProductionRuntime)

Sets the state of a work response.

Inputs	Data Type	Description
Response State	String	Specifies the state to set for the work response. Valid values are: <ul style="list-style-type: none"> • Ready • Completed • Holding • Aborted
Work Response	GUID	Specifies the ID of the work response you want to set the state for.

WriteResourceActualProperties (IProductionRuntime)

Writes properties to a specific resource actual.



Note:

In the Configure Method dialog box, click Next to select a resource actual and its properties you want to write. You can select all properties or a subset. Each property you select displays as an input parameter on the activity configuration panel.

Inputs	Data Type	Description
Actual ID	GUID	Specifies the value for each resource actual property to write to the resource actual.
Resource Type	Dependent on each parameter's data type	Specifies the resource type of the resource actual being written to. There are three options: <ul style="list-style-type: none"> • Material • Equipment • Personnel

WriteResourceRequirementProperties (IProductionRuntime)

Writes properties to a specific resource requirement.

**Note:**

In the Configure Method dialog box, click Next to select a resource actual and its properties you want to write. You can select all properties or a subset. Each property you select displays as an input parameter on the activity configuration panel.

Inputs	Data Type	Description
Requirement ID	GUID	<p>Specifies the value for the resource requirement to add the properties to. This is the ID that is returned when you create a segment response.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>
Resource Type	Dependent on each parameter's data type	<p>Specifies the resource type of the resource requirement being written to. There are three options:</p> <ul style="list-style-type: none"> • Material • Equipment • Personnel

WriteWorkData (IProductionRuntime)

Writes the values of a work data class to a specific segment response.

**Note:**

In the Configure Method dialog box, click Next to select a work data class and its properties you want to write. You can select all properties or a subset. Each property you select displays as an input parameter on the activity configuration panel.

Inputs	Data Type	Description
Segment Response ID	GUID	Specifies the value for the segment response. This is the ID that is returned when you create a segment response.
Work data input parameters	Dependent on each parameter's data type	Specifies the value for each work data property to write to the segment response.

WriteWorkParameters (IProductionRuntime)

Writes the value of specified work definition segment parameters to a process segment.



Note:

In the **Configure Method** dialog box, click **Next** to select a work data class and its properties you want to write. You can select all properties or a subset. Each property you select displays as an input parameter on the activity configuration panel.

Inputs	Data Type	Description
Requirement ID	String	Specifies the ID of the segment requirement to write the parameter values to.
Work data input parameters	Dependent on each parameter's data type	Specifies the value for each work definition segment parameter to write to the process segment.

IProficiencyHistorian

IProficiencyHistorian allows you to manage Historian servers, including reading from and writing to Historian collectors and tags.

On the activity configuration panel, you can use a *bind* or *expression* to define your input and output parameters.

Subtopics in this section describe each service method for this service interface.

AddCollector (IProficiencyHistorian)

Adds a new collector to an Historian server.

Inputs	Data Type	Description
collectorname	String	Specifies an Historian collector name for the server identified.
description	String	Specifies the description of the Historian collector.
historianAddress	DirectoryResource	Specifies the address of a GE Historian server.
type	CollectorType	<p>Specifies the Historian tag data type.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • ihInterfaceUndefined • ihIFix • ihRandom • ihOPC • ihFile • ihIFixLabData • ihManualEntry • ihOther • ihCalcEngine • ihServerToServer • ihPI • ihOPCAE • ihCIMPE • ihPIDistributor • ihCIMME



Note:

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description	
Response	Indicates the response based on the method's input values.	

AddTag (IProficyHistorian)

Adds a new tag to an Historian server.

Inputs	Data Type	Description
collectioninterval	Int32	Specifies an Historian tag collection interval in milliseconds.
collectorname	String	Specifies an Historian collector name associated with the tag identified.
description	String	Specifies the description of the tag identified.
historianAddress	DirectoryResource	Specifies the address of a GE Historian server.
sourceaddress	String	Specifies the source address of the tag identified.
tagName	String	Specifies a GE Historian tag name for the server identified.
type	HistorianTypeCode	<p>Specifies the Historian tag data types.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • ihDataTypeUndefined • ihScaled • ihFloat • ihDoubleFloat • ihInteger • ihDoubleInteger • ihFixedString • ihVariableString

Inputs	Data Type	Description
		<ul style="list-style-type: none"> • ihBlob • ihTime • ihMaxDataType

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description
Response	Indicates the response based on the method's input values.

CreateDataSourceLink (IProficiencyHistorian)

Creates a data source link to an Historian tag, which can be applied to an equipment property.

Inputs	Data Type	Description
historianAddress	DirectoryResource	Specifies the address of a GE Historian server.
linkType	LinkType	Specifies the type of data source link to retrieve. Valid values are: <ul style="list-style-type: none"> • CurrentValue • Historical
tagName	String	Specifies a GE Historian tag name for the server identified.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description	
DataAccessEndPoint		Indicates the link to an external data item for a specified tag.

DeleteCollector (IProficyHistorian)

Deletes a collector from an Historian server.

Inputs	Data Type	Description
collectorname	String	Specifies an Historian collector name for the server identified.
historianAddress	DirectoryResource	Specifies the address of a GE Historian server.

DeleteTag (IProficyHistorian)

Deletes a tag from an Historian server.

Inputs	Data Type	Description
historianAddress	DirectoryResource	Specifies the address of a GE Historian server.
tagName	String	Specifies a GE Historian tag name for the server identified.

Outputs	Description	
Response		Indicates the response based on the method's input values.

GetCollectors (IProficyHistorian)

Retrieves the name of the collectors belonging to an Historian server.

Inputs	Data Type	Description
historianAddress	DirectoryResource	Specifies the address of a GE Historian server.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description
String[]	Indicates the name of the Historian server collectors.

GetTagAttributes (IProficyHistorian)

Retrieves the attributes of the specified Historian tag.

Inputs	Data Type	Description
historianAddress	DirectoryResource	Specifies the Historian server that contains the tag whose attributes you want to get.
tagName	String	Specifies the tag name whose attributes you want to get.

SectionTitle

Outputs	Description
TagAttributes	Returns the attributes of the selected Historian tag.

Retrieve and configure Historian tag attributes

This procedure describes how to read from and configure attributes for a tag on an Historian server.

Procedure

1. Add a workflow that will be used to configure the attributes of an Historian tag.
2. In the navigator, click **Workflow > Workflows**.
3. In the Workflows list, select the resource you want to work with.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.

6. Insert an activity:

◦ At the workflow level:

- From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

◦ At the subprocess level:

- Double-click the local subprocess you want to add the activity to.
- From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

7. On the activity configuration panel, click **Configure Method**.

The **Configure Method** dialog box appears, displaying a list of the available call method service providers.

8. Double-click **IProficyHistorian** to display the list of available call methods.9. Click **GetTagAttributes**, and then click **Finish**.

10. Configure the call method as follows:

- a. In the **Historian Address** field, click the ellipsis button, and then select the Historian server with the tag whose attributes you want to retrieve.
- b. In the **Tag Name** field, enter the name of the Historian tag whose attributes you want to retrieve.

11. For each tag attribute you want to change, insert a Write activity into the workflow, after the call method, and then configure it as follows:

- a. On the activity configuration panel, click the ellipsis button.
The **Select Variable** dialog box appears.
- b. Expand the call method you created in step 9 and, then select the Return Value.
- c. Click **OK**.
- d. On the activity configuration panel, enter a value for the attribute or bind the value to another attribute.

12. Repeat steps 7 through 11.

13. Click **SetTagAttributes**, and then click **Finish**.

14. Configure the call method as follows:

- a. In the **Historian Address** field, click the ellipsis button, and then select the same Historian server you selected in step 10a.
 - b. In the **Tag Attributes** field, click the arrow, and then select **Bind**.
The **Select Variable** dialog box appears.
 - c. Expand the call method you created in step 9, and then select the Return Value output parameter.
 - d. In the **Tag Name** field, enter the name of the same Historian tag you specified in step 10b.
 - e. Click **OK**.
15. Click **Save**.
 16. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

SetTagAttributes (IProficyHistorian)

Sets the attributes of the specified Historian tag.



Note:

- We recommend that you use a Write activity to change the tag attributes before setting them. For more information, see [Configuring Historian Tag Attributes](#).

Inputs	Data Type	Description
historianAddress	DirectoryResource	Specifies the Historian server that contains the tag whose attributes you want to configure.
tagAttributes	TagAttributes	Specifies the attributes whose values you want to configure. We recommend that you modify only the following tag attributes for proper server-to-server tag setup for data collection:

Inputs	Data Type	Description
		<ul style="list-style-type: none"> • CollectorName • CollectorType • SourceAddress • CalculationTriggers • CollectionType • CollectionInterval • DataType • StoreMilliseconds <p>In the Configure Method dialog box, click Next to specify the tag attribute you want to set.</p>
tagName	String	Specifies the name of the tag whose attributes you want to configure.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

This method returns no data.

TagExists (IProficyHistorian)

Tests whether or not a tag exists on an Historian server.

Inputs	Data Type	Description
historianAddress	DirectoryResource	Specifies the address of a GE Historian server.
tagName	String	Specifies a GE Historian tag name for the server identified.

Outputs	Description	
Boolean	Indicates True or False based on the method's input values.	

IPropertyGroup

IPropertyGroup allows you to read from or write to the properties of a resource.

On the activity configuration panel, you can use a *bind* or *expression* to define your input and output parameters.

Subtopics in this section describe each service method for this service interface.

ReadPropertyGroup (IPropertyGroup)

Reads property values of a resource based on property names of the property group.

Inputs	Data Type	Description
Property Group	DirectoryResource	<p>Specifies the name of the property group with the properties you want to read.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>
Resource	DirectoryResource	<p>Specifies the resource whose properties you want to read.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>

Outputs	Description	
PropertyCollection		<p>Returns the property names and values as a collection.</p> <p>A property value returns null if the data type of the property group property is incompatible with the property of the resource or if the property does not exist.</p>

Read from a property group

This procedure describes how to read property values of a resource based on property names of the property group.

Before you begin

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.
5. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.



Note:

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
6. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.

7. Double-click **IPropertyGroup** to display the list of available call methods.
8. Click **ReadPropertyGroup**, and then click **Finish**.
9. Configure the call method as follows:
 - a. In the **Property Group** field, click the ellipsis button, and then select the property group you want to read.

**Note:**

The call method uses this property group to do batch reads of properties pertaining to a specified resource.

- b. In the **Resource** field, click the ellipsis button, and then select the resource whose properties you want to read.
10. Click **Save**.
11. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

WritePropertyGroup (IPropertyGroup)

Writes property values of a resource based on property names of the property group.

**Note:**

In the Configure Method dialog box, click Next to specify the property group to write to. After selecting the property group, you can select the properties you want to write.

Inputs	Data Type	Description
Property Group	DirectoryResource	Specifies the name of the property group whose properties you want to write to.
PropertyGroup.Property	Dependent on each parameter's data type	Specifies the value to write for the property. <div data-bbox="1036 1667 1086 1717" data-label="Image"> </div> <div data-bbox="1096 1677 1175 1709" data-label="Section-Header">Note:</div> <div data-bbox="1096 1719 1391 1848" data-label="Text"> <p>Properties are available for selection only after you select a property</p> </div>

Inputs	Data Type	Description
		 group on the Configure Methods dialog box.
Resource	DirectoryResource	<p>Specifies the resource with the properties you want to write to.</p> <p>In the Inputs section of the activity configuration panel, click the ellipsis button to open the Universal Browser to select the resource you want to use.</p>

Outputs	Description	
PropertyCollection		<p>Returns the property names and values as a collection.</p> <p>The property collection returned will contain any failed writes.</p> <p>A failed write can occur if the data type of the property group property is incompatible with the data type of the resource property or if the property of the property group does not exist for the resource.</p>

Write to a property group

Before you begin

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. In the navigator, click **Workflow > Workflows**.
2. In the Workflows list, select the resource you want to work with.
3. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
4. Click the **Workflow Editor** tab.

5. Insert an activity:

◦ At the workflow level:

- From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

◦ At the subprocess level:

- Double-click the local subprocess you want to add the activity to.
- From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

6. On the activity configuration panel, click **Configure Method**.

The **Configure Method** dialog box appears, displaying a list of the available call method service providers.

7. Double-click **IPropertyGroup** to display the list of available call methods.8. Click **WritePropertyGroup**, and then click **Next**.9. Click **Select property group**, and then select the property group whose properties you want to write to.

The properties configured for the group are displayed.

10. Select the properties whose values you want to write, and then click **Finish**.

The selected properties are displayed on the activity configuration panel.

11. Configure the call method as follows:

- In the **Resource** field, click the ellipsis button, and then select the resource to apply the property group to.
- In each of the property fields, enter or select the value to write.

12. Click **Save**.13. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

IResourceLinking

IResourceLinking allows you to work with resource links for all types of resources. You can start, edit, end, and query links between resources in your system.

Using resource links, you can create a link between any two resources. This allows you to track all the links for a specific resource or for a specific time period. For example, you can create one link between

an engine and an aircraft for a specified time period, and a second link for the same engine and another aircraft for a different time period.

On the activity configuration panel, you can use a *bind* or *expression* to define your input and output parameters.

Subtopics in this section describe each service method for this service interface.

AlterResourceLink (IResourceLinking)

Modifies an existing user-defined resource link.

Inputs	Data Type	Description
End Time	SateTime	<p>(Optional) Specifies a new end time for the link. If you leave this field blank, then the resource link's end time does not change.</p> <div data-bbox="1024 953 1419 1535" style="border: 1px solid #0070c0; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note: If the resource link already has an end time defined, you can modify it to an earlier or later date and/or time. If the resource link does not have a defined end time, you can set it here. In either case, the end time must be later than the start time.</p> </div>
Resource Link ID	GUID	Specifies the unique ID of the resource link you want to work with.
Resource Link Name	String	(Optional) Specifies the new name for the resource link. If this field is blank, then the resource link name does not change.

Inputs	Data Type	Description
Start Time	DateTime	(Optional) Specifies the new start time for the resource link. If you leave this field is blank, then the resource link's start time does not change.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

EndResourceLink (IResourceLinking)

Ends a user-defined resource link.

Inputs	Data Type	Description
End Time	DateTime	(Optional) Specifies the time that the resource link will end. If you leave this field is blank, then the resource link will end at the current date and time when the call method runs. <div data-bbox="1024 1283 1419 1724" style="border: 1px solid #00a0c0; border-radius: 10px; padding: 10px; margin-top: 10px;">  <p>Note: The end time must be later than the start time. If a resource link already has an end time configured for it, you can only configure a new end time that is earlier than the configured time.</p> </div>
Resource Link ID	GUID	Specifies the unique ID of the resource link you want to work with.

QueryCurrentResourceLinks (IResourceLinking)

Returns all current resource links for the specified source, target, and resource link name.



Note:

- You must provide a name, source, or target for the query, If you do not specify at least one of these parameters, the workflow will terminate with an error message stating that one of these parameters must be provided.
- If you specify only the resource link name, the information for all resource links with that name is returned.

Inputs	Data Type	Description
Resource Link Name	String	(Optional) Specifies the name of the resource link you want to query.
Source Address	DirectoryResource	(Optional) Specifies the source resource for the resource link.
Target Address	DirectoryResource	(Optional) Specifies the target resource for the resource link.



Note:

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description
ResourceLinkValues []	Returns the information for the specified query as an array.

Modify resource links using a workflow

This procedure describes how to change one or more attributes of a group of resource links with the same name.

Before you begin

This procedure can also be used to make a change to all resource links with the same source and/or target resources, or any combination of the input parameters used in the `QueryCurrentResourceLinks` and `QueryResourceLinksByTime` call methods.

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. Add a workflow to change one or more resource link attributes.
2. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
3. Double-click **IResourceLinking** to display the list of available call methods.
4. Click **QueryCurrentResourceLinks**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
5. In the **Resource Link Name** field, enter the name of the resource links you want to retrieve.
6. Insert a For Each activity, and then configure it as follows:
 - a. In the **Collection** field, click the arrow, and then select **Bind**.
The **Select Variable** dialog box appears.
 - b. Expand the call method you created in step 4, and then select the **Return Value** output parameter.
This binds the For Each activity to the output of the `QueryCurrentResourceLinks` call method.
 - c. Click **OK**.
7. Insert a Call Method activity inside the For Each activity.
8. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
9. Double-click **IResourceLinking** to display the list of available call methods.
10. Click **AlterResourceLink**, and then click **Finish**.
11. Configure the call method as follows:

- a. If applicable, in the **End Time** field, click the first arrow, and then select a new date and time for the resource link to end.



Note:

The end time must be later than the start time. If the resource link does not have an end time configured, you can select any date and/or time that occurs after the start time. If a resource link already has an end time configured for it, you can only configure a new end time that is earlier than the configured time.

- b. In the **Resource Link ID** field, click the arrow, and then select **Bind**. The **Select Variable** dialog box appears.
- c. Expand the ForEach activity you configured in step 6, expand the **Properties** list, and then expand the **Current Item** list.
- d. Select the **ResourceLinkID** property, and then click **OK**.
- e. If applicable, in the **Resource Link Name** field, enter a new name for the resource links returned by the **QueryCurrentResourceLinks** call method.
- f. If applicable, in the **Start Time** field, click the first arrow, and then select a new date and/or time for the resource link to begin.



Note:

The new start time can be earlier or later than the start time for the specified resource links as long as the start time has not occurred yet.

12. Click **Save**.
13. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

View resource links in a form

This procedure describes how to view your resource links using a form in a workflow.

Before you begin

Activities can be inserted at both the workflow and subprocess levels.

Procedure

1. Create a form and configure it as follows:
 - a. Add a control to use to view the resource link information.
 - b. In the **Forms Designer**, click the **Parameters** tab, add an input parameter to the form, and then configure the properties required.
 - c. Bind the **ItemSource** property of the control to an attribute of the input parameter, and then click **Save**.
2. Create a workflow that will be used to view resource links.
3. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
4. Double-click **IResourceLinking** to display the list of available call methods.
5. Click **QueryCurrentResourceLinks**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
6. Configure the query.

To query by...	Then...
resource link name	In the Resource Link Name field, enter the name of the resource links. This query returns all resource links with this name.
source resource	In the Source Address field, click the ellipsis button and select the source resource. The query returns all resource links with this source resource.
target resource	In the Target Address field, click the ellipsis button and select the target resource. The query returns all resource links with this target resource.

7. Insert a form activity after the call method, and then configure it as follows:
 - a. In the **Form** field, click the ellipsis button, and then select the form you created in step 1.
 - b. In the **InputParameter** field, click the arrow, and then select **Bind**.
The **Select Variable** dialog box appears.

- c. Expand the inserted call method, expand the output parameters, and then select the **Return Value** of the call method.
 - d. Click **OK**.
8. Click **Save**.
 9. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.

QueryResourceLinkByTime (IResourceLinking)

Returns all resource links with the specified source, target, and resource link name that exist within the specified time range.



Note:

- You must provide a name, source, or target for the query. If you do not specify at least one of these parameters, the workflow will terminate with an error message stating that one of these parameters must be provided.

Inputs	Data Type	Description
End Time	DateTime	(Optional) Specifies the date and time that the resource link is set to end. If you leave this field and the Start Time field blank, then the search will return all resource links with the defined criteria. If this field is set to the same date and time as the start time, then the query will search only on that exact time.
Resource Link Name	String	(Optional) Specifies the name of the resource link you want to query.
Source Address	DirectoryResource	(Optional) Specifies the source resource for the resource link.

Inputs	Data Type	Description
Start Time	DateTime	(Optional) Specifies the date and time that the resource link is set to start. If you leave this field and the End Time field blank, then the search will return all references with the defined criteria. If this field is set to the same date and time as the end time, then the query will search only on that exact time.
Target Address	DirectoryResource	(Optional) Specifies the target resource for the resource link.

**Note:**

In the **Configure Method** dialog box, click **Next** to select a resource. Otherwise, in the **Inputs** section of the activity configuration panel, click the ellipsis button to select a resource.

Outputs	Description
ResourceLinkValues []	Returns the information for the specified query as an array.

StartResourceLink (IResourceLinking)

Starts a user-defined resource link between two resources.

Inputs	Data Type	Description
Resource Link Name	String	Specifies a name for the resource link. <div data-bbox="1036 1667 1086 1717" data-label="Image"> </div> <div data-bbox="1096 1680 1175 1711" data-label="Section-Header">Note:</div> <div data-bbox="1096 1722 1391 1890" data-label="Text"> <p>You can create multiple resource links with the same name; however, the source and/or tar-</p> </div>

Inputs	Data Type	Description
		 get resources must be unique for each resource link.
Source Address	DirectoryResource	Specifies the source resource for the resource link.
Start Time	DateTime	(Optional) Specifies the start date and time for the resource link.  Note: If the Start Time field is blank, the start time defaults to the current date and time when the call method runs.
Target Address	DirectoryResource	Specifies the target resource for the resource link.

Outputs	Description
GUID	The unique ID of the new resource link.

Create resource links using a workflow

Before you begin

Activities can be inserted at both the workflow and subprocess levels.

About this task

This procedure describes how to change one or more attributes of a group of resource links with the same name.

Procedure

1. Add a workflow to create resource links.
2. In the navigator, click **Workflow > Workflows**.

3. In the Workflows list, select the resource you want to work with.
4. In the **Displays** panel, click **Workflow Management > Workflow Editor**.
5. Click the **Workflow Editor** tab.
6. Insert an activity:
 - At the workflow level:
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.

**Note:**

Activities can exist only between the presubprocess and postsubprocess.

- At the subprocess level:
 - Double-click the local subprocess you want to add the activity to.
 - From the **Standard Activities** panel, click and drag the activity you want to insert to a drop point.
7. On the activity configuration panel, click **Configure Method**.
The **Configure Method** dialog box appears, displaying a list of the available call method service providers.
 8. Double-click **IResourceLinking** to display the list of available call methods.
 9. Click **StartResourceLink**, and then click **Finish**.
The input parameters appear in the activity configuration panel.
 10. Configure the call method as follows:

- a. In the **Resource Link Name** field, enter a display name for the resource link.
- b. In the **Source Address** field, click the ellipsis button and select the source resource.
- c. If applicable, in the **Start Time** field, click the first arrow, and then select a date and time for the resource link to begin.

**Note:**

If the **Start Time** field is left blank, the start time defaults to the current date and time when the call method runs.

- d. In the **Target Address** field, click the ellipsis button and select the target resource.
- e. Click **Save**.

11. Repeat steps 5-6 to create as many StartResourceLink call methods as required. Use the same resource link name for each call method, but ensure that each has different source and/or target addresses.



Important:

All resource links with the same name must have different source and/or target resources. If you create a StartResourceLink call method that duplicates an existing call method, it is ignored.

12. Click **Save**.
13. Run the workflow, or, on the **Test/Debug** tab, test the workflow to ensure that your input parameters are configured correctly.
The resource links are created and can be viewed in a form.

iFix Service Provider

iFIX Service Provider

The iFIX Service Provider allows you to access iFIX tags and get active alarms in Workflow.

The iFIX Service Provider can be downloaded from the GE Technical Support web site, and is installed with an existing Workflow 2.1 application.

Chapter 14. Operator Task List in the Workflow Client

Task List Overview

Using the Task List

When a workflow is running in the Task List, it is ready for operators to use it.

The Task List

The Task List is an interactive display that provides users with information regarding task and task steps of a running workflow.

The Task List allows a user to view or perform a variety of workflow tasks, as well as to monitor the progress of a these tasks. Within this list, an operator can view one or more tasks awaiting user attention. The displayed tasks are determined by the logged in user ID, user role, and the equipment location of the physical computing resource presenting the client application. This Task List is a component that manages the interaction between operators and the workflow *execution engine*. It enables workflows that require human interaction to progress.

The Task List also enforces authorization of specific Task List actions through the use of Workflow security and role definitions. For example, the logged in user may have the ability to start a task but not skip a task, or delegate the task to someone else.



Tip:

The Workflow Task List can be integrated into other GE HMI applications. For more information, see *Task Controls in GE HMIs*.

General Operating Environment

General Operating Environment

The Task List has several unique features to help you navigate through, and perform, your tasks.



Note:

If you are using an Android device for working with the Mobile Task List, we recommend that you use Chrome version 31 or newer.

Title Bar

To log out of the Task List, tap your user name.

Icon	Description
	Home
	Equipment Asset List
	Refresh
	Data Alert
	Updates

Navigation Bar

The following icons give you information about the state of your Task List.

Icon	Description
	Tasks
	Tasks Available
	Task Expired
	Task Selected

If you have a large number of tasks in your Task List, you can navigate through the pages using the arrow buttons that are located at the top of the list.

Icon	Description
	Refresh
	Moves to the next section of the list.

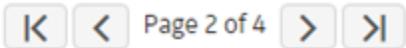
Icon	Description
	Skips to the last section of the list.
	Returns to previous section of the list.
	Skips to the first section of the list.

Action Bar

You can expand or collapse the Action Bar.

Icon	Description
	Action Bar
	Help
	Task
	Equipment Context
	Filters

Action Bar

Icon	Description
	Task Pages
	Filters
	Tasks
	Start Task

Task List Functions

Icon	Description
	Actions
	Documents
	Work Instructions
	Forms
	Manual Start

Task List Functions

Icon	Description
	Actions
	Documents
	Work Instructions
	Forms
	Manual Start

Select a task view

Procedure

1. Log on to the Task List.
2. If applicable, select the view you want to work in.



Note:

The default view is the *Tasks* tab.

Click	To view
Tasks	workflow tasks currently being executed.
Task History	completed tasks.

3. If required, select a task.

Sort task information

Procedure

1. Log on to the Task List.
2. If applicable, select the view you want to work in.



Note:

The default view is the **Tasks** tab.

Click...	To view...
Tasks	workflow tasks currently being executed.
Task History	completed tasks.

3. If required, select a task.
4. Select a view.

In **Tasks** view:

Click...	To sort by the task...
Title	name
Duration	start date and time
Expiry	expiration countdown in days (hours/minutes/seconds)
Priority	prioritization

In **Task History** view:

Click...	To sort by the task...
Title	name

Click...	To sort by the task...
Start Time	start date and time
Completed Time	end date and time
Expiry	expiration countdown in days (hours/minutes/seconds)
Priority	prioritization
State	task run time state

Increase text size

You can increase the text size for work instructions and forms.

Before you begin

When the zoom feature is available, you can adjust the size of the words on the screen. This slide zoom allows you to increase the scale by five to ten times the original size. The settings of these adjustments are saved when you log out.



Note:

You can only zoom WPF input forms.

Procedure

1. Log on to the Task List.
2. If applicable, select the view you want to work in.



Note:

The default view is the **Tasks** tab.

Click...	To view...
Tasks	workflow tasks currently being executed.
Task History	completed tasks.

3. If required, select a task.
4. Click **Zoom**.
5. Move the slider:
 - up to increase the text size, or
 - down to decrease the text size.

Use the context navigator to move between tasks/task steps

Procedure

1. Log on to the Task List.
2. If applicable, select the view you want to work in.



Note:

The default view is the **Tasks** tab.

Click...	To view...
Tasks	workflow tasks currently being executed.
Task History	completed tasks.

3. If required, select a task.
4. In the context navigator, do one of the following:
 - a. To view more information about a task or task step, hover over a bar.
 - b. To select a task or task step, click the task or task step bar.

Change layout mode

Procedure

1. Log on to the Task List.
2. If applicable, select the view you want to work in.



Note:

The default view is the **Tasks** tab.

Click...	To view...
Tasks	workflow tasks currently being executed.
Task History	completed tasks.

3. If required, select a task.
4. On the control bar, click:
 - Vertical Layout Mode, or
 - Horizontal Layout Mode

Change font size

Procedure

1. Log on to the Task List.
2. If applicable, select the view you want to work in.



Note:

The default view is the **Tasks** tab.

Click...	To view...
Tasks	workflow tasks currently being executed.
Task History	completed tasks.

3. If required, select a task.
4. On the control bar, for small, medium, or large text, click the appropriate button: .



Note:

The font sizes change in the task bar, task grid, task headings, and Task Steps sections only.

Task States

States indicate what is happening at a certain point within a running workflow.

States can be viewed in the Task History tab, a report, or in the Task Steps panel in the Task List.

States

The following tables display the different states that can be presented during run time, or through the history, or reporting:

Task States

State	Description
Running	in execution.
Cancelled	Cancelled by the operator or system.
Completed	Completed as required.

State	Description
Stopped	Stopped by the operator.

Task Step States

State	Description
Waiting for Start	A manual start is required.
Running	In execution.
Cancelled	Cancelled by the operator or system.
Skipped	Not executed due to a jump action or a condition branch.
Completed	Completed as required.
Not Started	Task step has not been reached in the execution cycle yet.

States Displayed Visually

The states are displayed in color and graphically during run time to provide a visual clue for quick reference.

Color

The following table provides a description of the state colors:

Color	The task or task step is...
Green	complete
Blue	not started
Red	expired
Grey	assigned to another operator
White or Clear	currently selected

Icons

The following table provides a description of the state icons:

Icon	The task or task step is...
	Waiting for Start
	Running
	Cancelled
	Skipped
	Completed
	Not Started
	Stopped

Related information

[Filter task step states \(on page 2053\)](#)

Task Bar User Interface

The Task Bar provides an overview of the currently running workflow task. Within these tasks are task steps that provide more specific information about a step, and display any user interactions that are required.

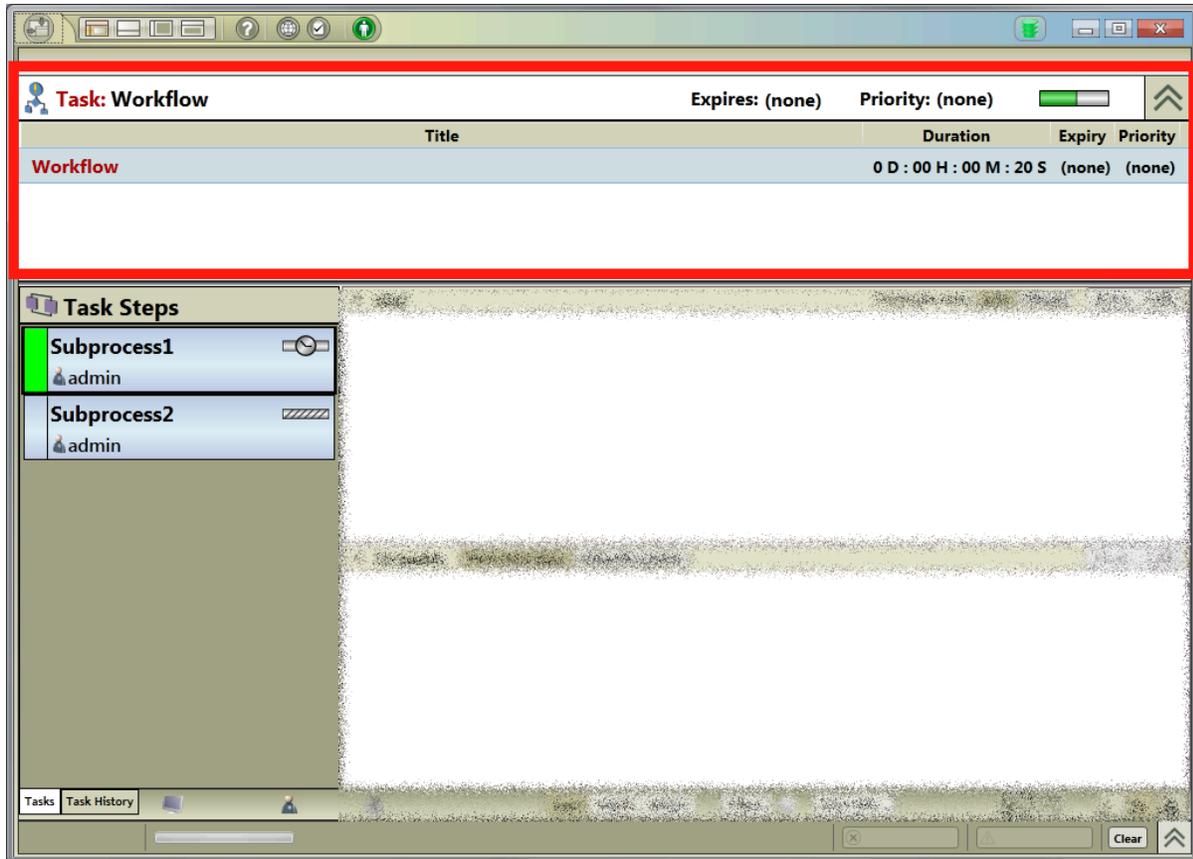
Attributes

Attribute	Value	Description
Task Name	User-defined; Preconfigured	<p>The given name for a workflow task.</p> <div style="border: 1px solid #00a0e3; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note: This name can be different from the task step name.</p> </div>
Expires	User-defined; Preconfigured	The scheduled duration, from the soonest to the latest time. Task

Attribute	Value	Description
		expirations take precedence over task step expirations. Expired tasks can still be performed.
Priority	User-defined; Preconfigured; Numeric	The preset relevance within the workflow task configuration, from the most important to the least important.
Status bar	System-defined	Indicates the running state of the task. For more information, see <i>Task States</i> .
	N/A	Click this button to open and close the task grid.

User Interface

The task bar is located at the top of the Task List.



Related reference

[Task States \(on page 2014\)](#)

Task Grid User Interface

The task grid provides an overview of tasks.

Attributes

There are two ways to view the task grid:

- **Tasks** tab, which is an overview of currently running tasks.
- **Task History** tab, which is an overview of tasks within a selected time frame.

User Interface

Attributes in the task grid can be sorted by clicking their respective headings. You can also open or close the grid to increase your workspace.

The screenshot shows a window titled "Task: Workflow" with a status bar indicating "Expires: (none)" and "Priority: (none)". Below the status bar is a table with the following data:

Title	Duration	Expiry	Priority
Workflow	0 D : 00 H : 00 M : 32 S	(none)	(none)
Workflow	0 D : 00 H : 00 M : 17 S	(none)	(none)

Below the table is a section for "Subprocess2" with the user "admin". At the bottom, there is a context navigator showing three subprocesses: "Subprocess1" (green bar), "Subprocess2" (blue bar), and "Subprocess3" (blue bar). Each subprocess bar includes a user icon and the name "admin".

Context Navigator User Interface

The context navigator is a real-time control that allows users, simultaneously, to view and navigate all tasks and task steps associated with all currently running workflows.

Attributes

The workflow task navigator is situated between the task grid and Task Step panels; however, it is visible only if there is more than one workflow running. The task step context navigator appears only if there are three or more workflows running at the same time. They act as scroll bars, appearing and disappearing as tasks and task steps run. Users can scroll through the list by clicking different bars.

The size and color of the bars in the navigator change depending on how many task steps are running and the state they are in. Because these bars exist in real-time, the states change constantly. If you touch a bar, the large version of the task step displays so you can view the details of that step. When a bar flashes, it is waiting for operator interaction. Thus, it is a visual overview of production activities.

User Interface

The Task List displays two context navigators, one each for workflow tasks and task steps.

This screenshot is identical to the one above, showing the "Task: Workflow" window with the task list and subprocesses. A red rectangular box highlights the context navigator area at the bottom of the window, which contains the three subprocess bars: "Subprocess1", "Subprocess2", and "Subprocess3".

Task Steps User Interface

The Task Steps section offers you a visual display of the task steps being executed or that require execution.

Attributes

The task steps available to be viewed depend on the layout mode and user permissions of your Task List. If a workflow has any *Status Message activities* associated with it, this panel displays that information.

Each task step displays the following information:

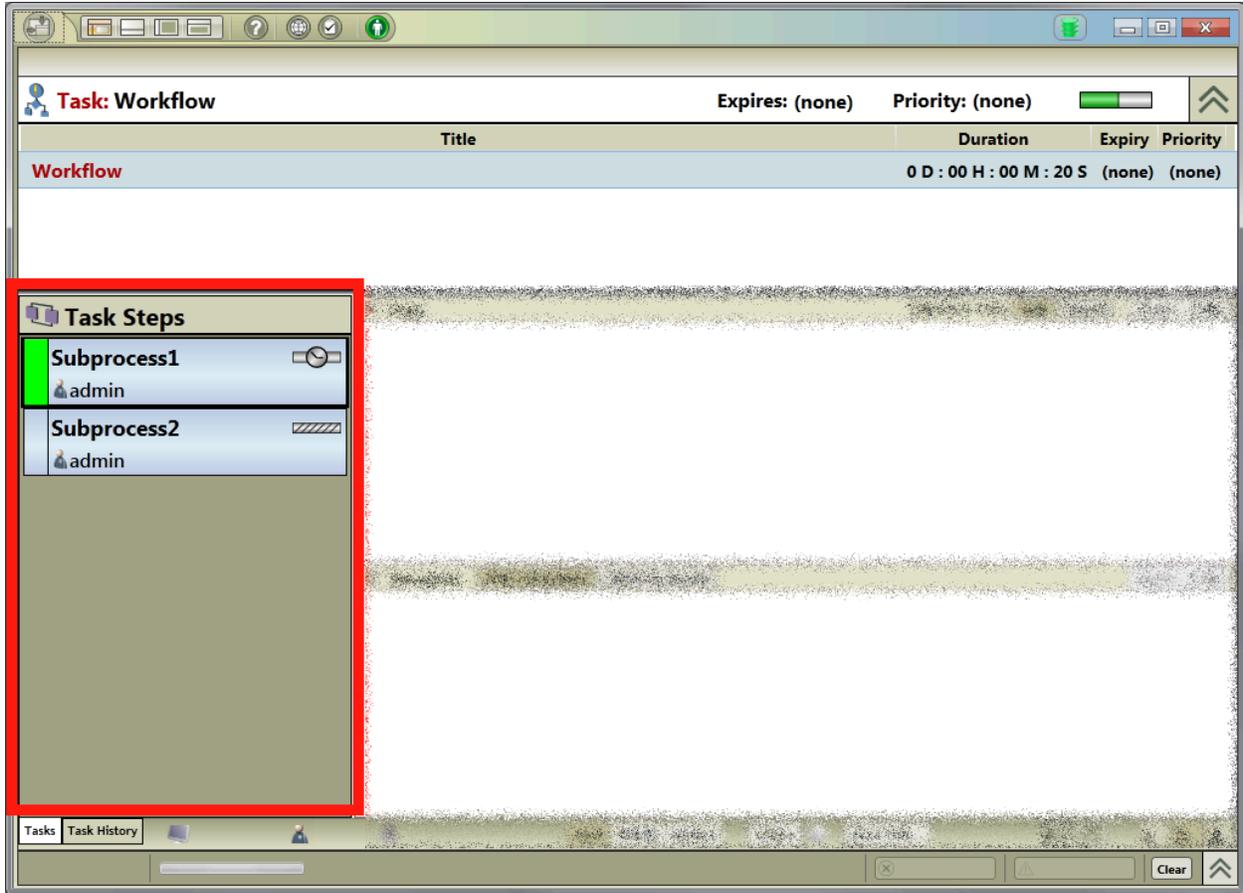
- name of task step
- state
- location the task step is assigned to

When a task step is started, the following information is also displayed:

- personnel the task step is assigned to

User Interface

Each task related to a workflow is listed in Task Steps.



Control Bar User Interface

The control bar at the bottom of the Task List allows you to navigate and perform various actions.

Attributes	Value	Description
Tasks tab	N/A	Default view for a running workflow.
Task History tab	N/A	Specifies all previously completed tasks and their task steps within a defined time range.
	N/A	Context button. Click to associate a user with a piece of equipment.
	User-defined; Preconfigured	Location and Personnel display. Specifies the location and the

Attributes	Value	Description
		user that a task is assigned to. This information is for display only and cannot be changed from this location; it is changed on the Actions tab.
Details button	System-defined	Provides information regarding a task and a task step in a list view on the Tasks tab.
Actions button	User-defined; Preconfigured	Specifies specific task operations when a workflow is running (Tasks tab) and when a workflow is completed (Task History tab).
Filters button	User-defined	Allows context sensitive filtering of task steps. When one or more filters are selected, the filter feature turns on, and the filter indicator turns green.
Start Task button	User-defined	Allows you to start a workflow schedule during the execution of a task or task step.
	User-defined	Vertical Layout Mode. Arranges the steps you are working on in a vertical list. These tasks scroll up and down, showing the previous, current, and next steps, as they are completed. You can also manually scroll the list by selecting previous and next steps, which automatically shifts the view and highlights the step selected.
	User-defined	Horizontal Layout Mode. Highlights the step you are working on to a maximum of three steps

Attributes	Value	Description
		<p>per screen. The current task is in the middle, and the easiest to view. These tasks continue to scroll from right to left, showing the previous, current, and next steps, as they are completed. You can also manually scroll the list by selecting previous and next steps, which automatically shifts the view and highlights the step selected.</p>
	User-defined	<p>Adjust the size of the text in the task bar, task grid, and Task Steps sections. You can adjust the sizes using a small, medium, and large template. The panel sizes adjust to accommodate the new text sizes, and these settings are saved at logout. These text sizes allow you to adjust your workspace depending on your requirements and to improve the usability of your workspace.</p>

General Task Information

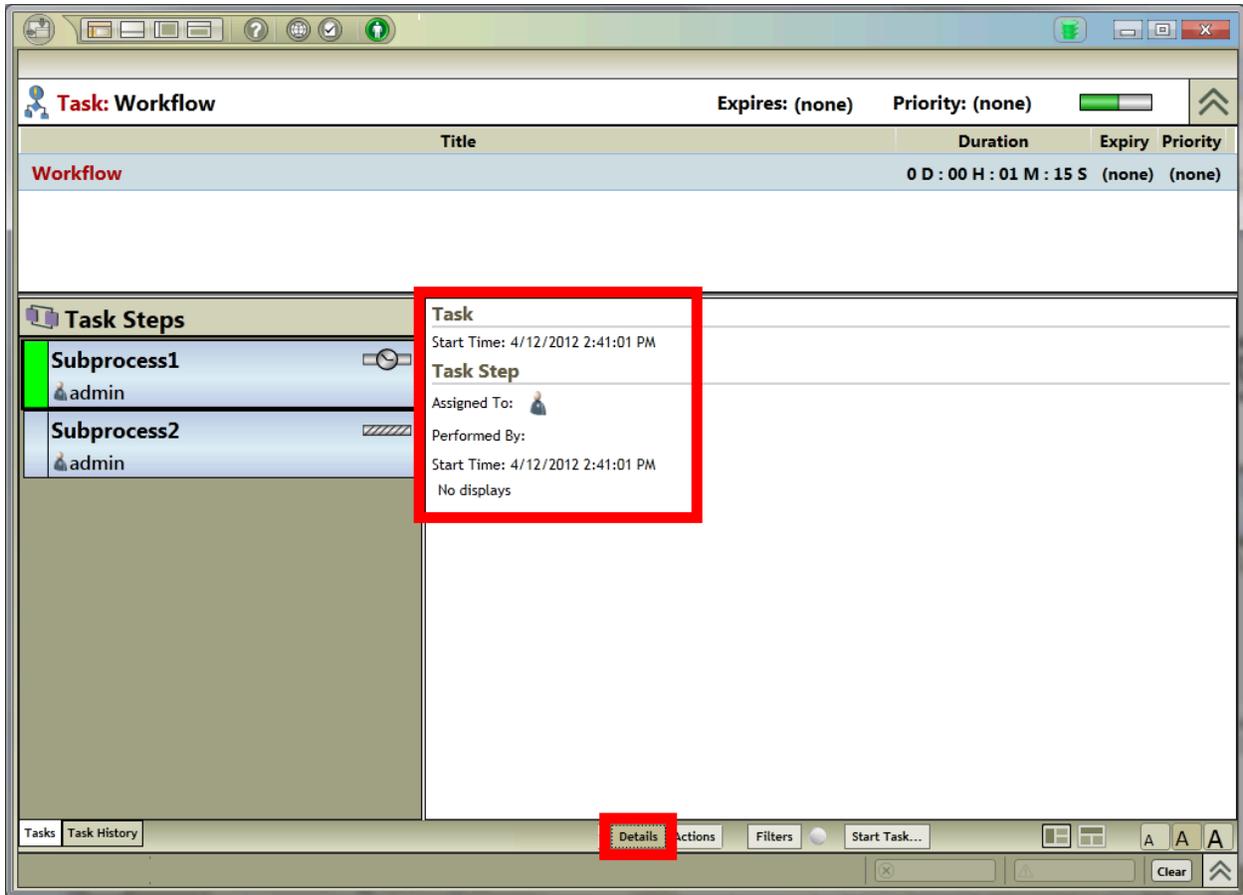
General Task Information

The Detail button provides information regarding a task and a task step in a list view, such as start/end times and personnel/location assignments.

It displays any information that may require user interaction, such as a manual start or a form submission for a task step.

User Interface

Click the **Details** button to see more information.



View task details

Procedure

1. Log on to the Task List.
2. If applicable, select the view you want to work in.



Note:

The default view is the **Tasks** tab.

Click...	To view...
Tasks	workflow tasks currently being executed.

Click...	To view...
Task History	completed tasks.

3. If required, select a task.

Equipment Context

Equipment Context Show Tasks for Equipment

You can manually set the Task List equipment context, which becomes associated with you as an operator. You must be logged in to a Task List to set the equipment context. You can manually set the Task List equipment context, which becomes associated with you as an operator. You must be logged in to a Task List to set the equipment context.

If no equipment context is linked to an operator, then the context defaults from a linked piece of equipment to the computer you are using. Otherwise, there will be no equipment context if there is no linked piece of equipment to the computer.

This feature is related to the regular Task List only, not the Web Task List or Mobile-sized Task List.



Note:

Setting the equipment context on one Task List does not affect user settings on other lists unless each list is manually refreshed.

Attribute	Description
No Equipment button <u>OR</u> 	Use this feature to open the Set Equipment Context browser, and then select a piece of equipment. You can continue navigating to the child equipment by clicking on entries in the Equipment list.
OK	Click this button to set the equipment context to the currently selected equipment.
Cancel	Click this button to return to the previous screen without selecting equipment.

Attribute	Description	
Clear	Click this button to clear all previously selected equipment settings.	

Setting the Equipment Context from the Task List

When you are in the Task List, you can add or remove an equipment context.

Attribute	Description	
	Tap the equipment context icon to open the Settings dialog box, where you can toggle the equipment context on or off.	
View Tasks by Equipment Context	In the Settings dialog box, specify a task's associated equipment context.	

Attribute	Description	
Show Task for Equipment	Select a piece of equipment and then slide the toggle switch on to display all tasks that are associated with a selected piece of equipment. Toggle off to hide this information.	

Setting an Asset as the Equipment Context

From the Assets list, you can go to the Task List to set the selected asset as the equipment context.

Icon	Description
	Tap the Tasks icon to view the tasks for the selected asset.

Set equipment context during run time

You can manually set the Task List equipment context, which becomes associated with you as an operator. You must be logged in to a Task List to set the equipment context.

Before you begin



Note:

All manual settings are overridden if a form in the Task List uses the **Set Equipment Context** client method to assign equipment.

Procedure

- To set the equipment context:

From the...	Click...
Web Task List or Task List	No Equipment button
Mobile-sized Task List	

The **Set Equipment Context** dialog box appears.

- Optional:** To remove previous equipment settings, click **Clear**.
- Optional:** To select nothing, click **Cancel**.
- Select the equipment that you want to set by clicking through the available options, and then click **OK**.

Forms View

Forms View

You can view and complete forms for the selected *task step*.

The Views button is the default view in the Tasks tab. From the regular and web task lists, you can view inputs as read-only from the Task History tab. To see a list of forms, go to the Input drop-down list.



Note:

HTML5 forms are not supported in the Web Task List.



Note:

You must have authorization to perform certain actions in the ActiveX Task List. If you cannot work in this area, you may not have the permission to do so, but you may still view the contents.

Web-based (HTML) Forms in the Task List

You can view forms that have been cancelled or submitted. If a form has been changed or deleted outside of the Task List, the form you view may not accurately reflect the original version.



Note:



If the **Form** icon is enabled, you can view completed forms.

Cancelled Forms

If a form has been canceled before it was submitted, you are able to view the initial form fields only, not any entries that an operator has made.

Submitted Forms

If a form has been submitted (that is, fully completed), then you are able to view the form's contents as it appeared during submission. You cannot, however, submit the form again.

Work with forms

You can view and complete forms for the selected *task step*.

Before you begin



Note:

Operator input may not be required for a *task step*.

Procedure

1. Log on to the Task List.
2. If applicable, select the view you want to work in.

**Note:**

The default view is the **Tasks** tab.

Click...	To view...
Tasks	workflow tsks currently being executed.
Task History	completed tasks.

3. If required, select a task.
4. Click **Views**.
5. Complete the form, and then click **Submit**.

**Note:**

For web-based forms, you must click the **Submit** button to submit your form. For all other forms, you can press `Enter` to submit your form.

Related information

[Sign forms electronically \(on page 2073\)](#)

Select multiple entries in a list box form

You can view and complete forms for the selected *task step*.

Procedure

1. Log on to the Task List.
2. If required, select a task.
3. Click **Views**.
4. From the list in the form, press the `CTRL` key, and then click each option that you want to select.
5. Click **Submit**.

The form panel collapses.

Manual Inputs

The Inputs section displays information that requires operator interaction for a task step, such as a manual start or a form submission. If a task is preconfigured to wait for an operator to start it, you must manually start a task step to progress the workflow.

If configured, you can view completed or acquired task forms. When a completed task step is selected, the last submitted form for that task step is shown. If an acquired task step is selected, the currently executing form is shown.

Input	Value	Description
Manual Start	Preconfigured	Requires operator interaction. The Start button advances the workflow.
Form input	User-defined; Preconfigured	<p>Requires operator interaction. Complete the form and submit it to advance the workflow. If applicable, cancel the form to advance the workflow; however, the Cancel button is available only if the user has permission to do so. You can also use the Status Message to get more information about your form.</p> <div data-bbox="574 743 1422 1058" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Tip: In the Mobile-sized Task List, you can use the arrow next to the Form button to display a selection of all available forms, both running and completed, and then select a form from the list. When there are no more task steps to perform, you will be notified and the <i>Start Task</i> button becomes available.</p> </div>
E-signature	User-defined; Preconfigured	<p>Requires operator interaction. A form may need to be signed electronically before it is submitted. Sign the form and submit it to advance the workflow.</p> <div data-bbox="574 1241 1422 1457" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: This feature is not available on the Mobile-sized Task List. You will receive an error message and the form will not load if you try to open a form that requires an e-signature.</p> </div>

Related reference

[Start Task \(on page 2065\)](#)

Input information manually for a task step

You can start a task step manually.

Before you begin

Depending on your access permissions and client configuration, you can select either My Task List or Task Management view.

Procedure

1. Log on to the Task List.
2. If applicable, select the view you want to work in.

 **Note:**
The default view is the **Tasks** tab.

Click...	To view...
Tasks	workflow tsks currently being executed.
Task History	completed tasks.

3. Operator may perform one of the following actions:
 - If a manual start is required, click **Start**.
 - If a form is required, complete the form, and then click **Sign**.

 **Note:**
Information displays in the *Views* panel only if operator interaction is required. The workflow may not progress until this requirement has been met.

Documents

Documents

You can view additional information that is helpful to complete a task or task step, such as web page content or a spreadsheet. If no documents are available, this feature is not available.

 **Tip:**
In the Mobile-sized Task List, you can use the arrow next to the **Documents** button to display a selection of all available linked documents and work instructions, both running and completed.

Attribute	Value/icon	Description
Linked Documents	User-defined; Preconfigured	<p>Read-only documents or web site URLs previously added to aid in the completion of a task step. If the original file document is updated, the updated version displays. Linked documents are available to users only after a task step starts running.</p> <div data-bbox="1027 653 1422 1146" style="border: 1px solid #00a0c0; border-radius: 10px; padding: 10px; background-color: #e6f2ff;"> <p> Note: Documents are embedded in this panel, and close automatically, unless your Microsoft Windows configuration does not allow this. Any errors opening a document are displayed in the Feedback Zone.</p> </div> <p>The following exceptions apply:</p> <ul style="list-style-type: none"> • Web Task List - To open documents in a new tab, in a new window, or in the native application for that file type, do the following: <ul style="list-style-type: none"> ◦ Set the URI for each link to an absolute path. ◦ Save documents in a folder on a server, giving the folder share permissions.

Attribute	Value/icon	Description
		<ul style="list-style-type: none"> • Mobile-sized Task List - To use embedded documents, you must <i>add a registry patch</i>.

View linked documents

You can view additional information that is helpful to complete a task or task step, such as web page content or a spreadsheet. If no documents are available, this feature is not available.

Before you begin



Tip:

- Mobile-sized Task List only: To view Microsoft Office 2007 and later documents inside an Internet Explorer browser window, *add a registry patch*.
- Web Task List only: To open embedded documents, set the URI for each link to an absolute path, save documents in a folder on a server, and give the folder share permissions.

Procedure

1. Log on to the Task List.
2. If applicable, select the view you want to work in.



Note:

The default view is the **Tasks** tab.

Click...	To view...
Tasks	workflow tsks currently being executed.
Task History	completed tasks.

3. If required, select a task.
4. Click the **Linked Documents** button.
5. If applicable, select one of the following options:



Note:

The name of task/task step is in parentheses.

For...	Select...
Task-level documents	Task
Task step documents	Task Step

6. Click the link for the document or web site you want to view.
7. If applicable, for embedded documents, click **Close** to collapse the information that you are viewing.

Operator Work Instructions

Operator Work Instructions

This section displays information that is helpful for you to complete a task, such as detailed instructions. If no instructions are available, this feature is not available.



Tip:

In the Mobile-sized Task List, you can use the arrow next to the **Documents** button to display a selection of all available linked documents and work instructions, both running and completed.

Attribute	Value/icon	Description
Work Instructions	User-defined; Preconfigured	Text instructions or graphical information previously added to aid in the completion of a task or task step. There are three options that may be visible when you select the work instructions button. If no instructions are available, this button will not be visible.

Attribute	Value/icon	Description
		<ul style="list-style-type: none"> • If instructions are available for both a <i>task</i> <u>and</u> a <i>task step</i>, then you can select the option related to the instructions that you want to view. • If instructions are available for only a <i>task</i> <u>or</u> a <i>task step</i>, then the instructions are automatically displayed for the appropriate level. • If no instructions are available, no information will be visible.

View work instructions

This section displays information that is helpful for you to complete a task, such as detailed instructions. If no instructions are available, this feature is not available.

Procedure

1. Log on to the Task List.
2. If applicable, select the view you want to work in.



Note:

The default view is the **Tasks** tab.

Click...	To view...
Tasks	workflow tsks currently being executed.
Task History	completed tasks.

3. If required, select a task.
4. Click the **Work Instructions** button.
5. If applicable, select one of the following options:

For...	Select...
Task-level documents	Task
Task step documents	Task Step

6. If applicable, for embedded documents, click **Close** to collapse the information that you are viewing.

Advanced Operator Tasks

Advanced Task Actions for Operators

Operators can perform advanced operations in a Task List.



Note:

Actions may or may not be available in all task lists. Click **Refresh** to view all relevant tasks in the task grid.

Available Actions

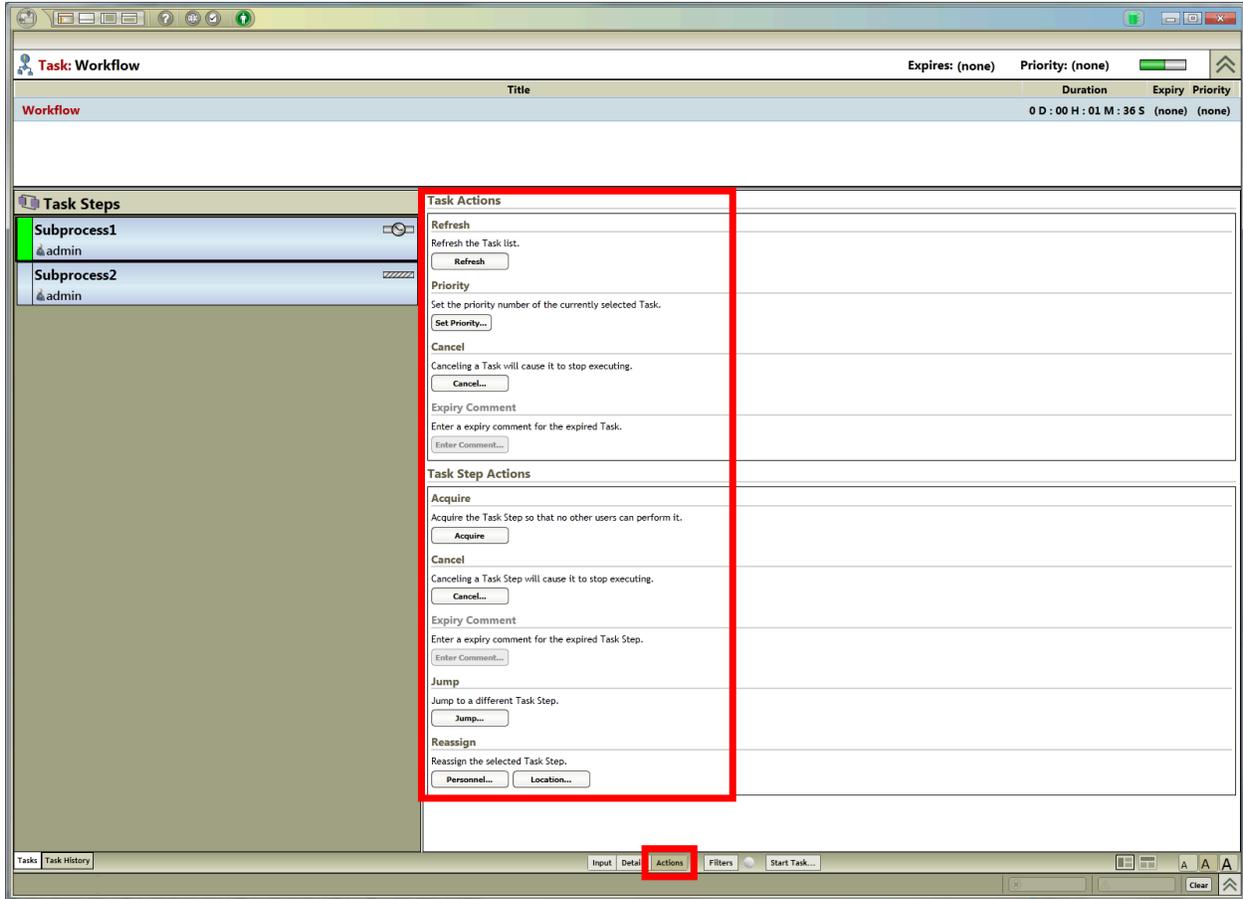
From the **Tasks** tab, you can perform advanced task or task step operations. If any of these actions require an electronic signature for verification, you will be prompted for a user name and password.

Historical Task Information

When a workflow is finished running, the related task information is available in **Task History**. Operators can enter expiry comments on completed or expired tasks and task steps.

User Interface

Click the **Actions** button to see more information.



Access the Actions button

Procedure

1. Log on to the Task List.
2. If applicable, select the view you want to work in.



Note:

The default view is the **Tasks** tab.

Click...	To view...
Tasks	workflow tsks currently being executed.
Task History	completed tasks.

3. If required, select a task.
4. Select the task that you want to work with.

5. Click the task step that you want to work with.
6. Click **Actions**.

Refresh tasks

Depending on your access permissions and client configuration, you can select either My Task List or Task Management view.

Before you begin



Note:

A workflow must be running to perform this action.

Procedure

1. Log on to the Task List.
2. If applicable, select the view you want to work in.



Note:

The default view is the **Tasks** tab.

Click...	To view...
Tasks	workflow tsks currently being executed.
Task History	completed tasks.

3. If required, select a task.
4. Select the task that you want to work with.
5. Click the task step that you want to work with.
6. Click **Actions**.
7. Click **Refresh**.

Add a comment to an expired task or task step

When a task or task step has expired, you can enter a statement explaining the reason for the expiration.

Before you begin

Depending on your access permissions and client configuration, you can select either My Task List or Task Management view.

**Note:**

Expiry comments are available only after the task or task step has expired.

Procedure

1. Log on to the Task List.
2. If applicable, select the view you want to work in.

**Note:**

The default view is the **Tasks** tab.

Click...	To view...
Tasks	workflow tsks currently being executed.
Task History	completed tasks.

3. Select the task that you want to work with.
4. click the tasts step that you want to work with.
5. Click **Actions**.
6. Click the appropriate **Enter Comment** button.

Location	Result
Task Actions	The Task Expiry Comment dialog box appears.
Task Step Actions	The Task Step Expiry Comment dialog box appears.

7. In the **Expiry Comment** field, enter a reason the task or task step expired.
8. Click **OK**.

Set a priority for a task or task step

You can apply a numeric value for a task. This priority setting does not affect the execution of the workflow; it is used only for sorting and filtering within the Task List.

Before you begin

Depending on your access permissions and client configuration, you can select either My Task List or Task Management view.



Note:

A workflow must be running to perform this action.

Procedure

1. Log on to the Task List.
2. If applicable, select the view you want to work in.



Note:

The default view is the **Tasks** tab.

Click...	To view...
Tasks	workflow tsks currently being executed.
Task History	completed tasks.

3. If required, select a task.
4. Select the task that you want to work with.
5. Click the task step that you want to work with.
6. Click **Actions**.
7. Click **Set Priority**.
8. In the **Priority** field, enter a value.



Note:

You can enter whole numbers only, such as 0, 1, 2, 3, and so on.

9. If an electronic signature is required, enter the appropriate information to authorize the action.
10. Click **OK**.

Cancel a task or task step

Operators can cancel an entire task or an individual task step in the Task List.

Before you begin

Depending on your access permissions and client configuration, you can select either My Task List or Task Management view.



Note:

A workflow must be running to perform this action.

Procedure

1. Log on to the Task List.
2. If applicable, select the view you want to work in.



Note:

The default view is the **Tasks** tab.

Click...	To view...
Tasks	workflow tsks currently being executed.
Task History	completed tasks.

3. Select the task that you want to work with.
4. Click the task step that you want to work with.
5. Click **Actions**.
6. Click the appropriate **Cancel** button.

Location	Result
Task Actions	Cancels the task and all task steps contained within the task.
Task Step Actions	Cancels only the selected task step, or completes the workflow.

7. In the **Cancel Reason** field, enter a reason for cancelling the task or task step.



Important:

This action cannot be reversed.

8. If an electronic signature is required, enter the appropriate information to authorize the action.
9. Click **OK**.

Acquire a task step to perform

You can acquire a task step when a workflow is running.

Before you begin

Depending on your access permissions and client configuration, you can select either My Task List or Task Management view.



Note:

A workflow must be running to perform this action.

Procedure

1. Log on to the Task List.
2. If applicable, select the view you want to work in.



Note:

The default view is the **Tasks** tab.

Click...	To view...
Tasks	workflow tsks currently being executed.
Task History	completed tasks.

3. Select the task that you want to work with.
4. Click the task step that you want to work with.
5. Click **Actions**.
6. Click **Acquire**.

Release a task step

You can release a task step from your list so that it can be acquired by another operator.

Before you begin

Depending on your access permissions and client configuration, you can select either My Task List or Task Management view.



Note:

A workflow must be running to perform this action.

Procedure

1. Log on to the Task List.
2. If applicable, select the view you want to work in.

**Note:**

The default view is the **Tasks** tab.

Click...	To view...
Tasks	workflow tsks currently being executed.
Task History	completed tasks.

3. Select the task that you want to work with.
4. Click the task step that you want to work with.
5. Click **Actions**.
6. Click **Release**.

Jump to a task step in the Task List

You can skip to, or jump to, a particular task in the Task List.

Before you begin

Depending on your access permissions and client configuration, you can select either My Task List or Task Management view.

**Note:**

To perform a jump, the Task List requires more than one task step.

Procedure

1. Log on to the Task List.
2. If applicable, select the view you want to work in.

**Note:**

The default view is the **Tasks** tab.

Click...	To view...
Tasks	workflow tsks currently being executed.
Task History	completed tasks.

3. Select the task that you want to work with.
4. Click the task step that you want to work with.

5. Click **Actions**.
6. In the **Task Step Actions** section, click **Jump**.
7. Select the task step, or set of steps, that you want to jump to.



Important:

When performing a jump action, currently executing task steps are cancelled. Jumping to a later task step in a task causes all uncompleted task steps before the jump target to be skipped.

The task steps that are available to jump to are preset, so not all steps may be available at all times.

8. If an electronic signature is required, enter the appropriate information to authorize the action.
9. Click **OK**.

Reassign personnel for a task step

You can reassign a task step to a different person, as required, in the Task List.

Before you begin



Note:

A workflow must be running to perform this action.

Procedure

1. Log on to the Task List.
2. If applicable, select the view you want to work in.



Note:

The default view is the **Tasks** tab.

Click...	To view...
Tasks	workflow tsks currently being executed.
Task History	completed tasks.

3. Select the task that you want to work with.
4. Click the task step that you want to work with.
5. Click **Actions**.

6. In the **Task Step Actions** section, click **Personnel**.
The **Universal Browser** appears.
7. Select the new user you want to assign this task step to.
8. If an electronic signature is required, enter the appropriate information to authorize the action.
9. Click **OK**.

Reassign a location for a task step

You can reassign a task step to different location or piece of equipment, as required, in the Task List.

Before you begin



Note:

A workflow must be running to perform this action.

Procedure

1. Log on to the Task List.
2. If applicable, select the view you want to work in.



Note:

The default view is the **Tasks** tab.

Click...	To view...
Tasks	workflow tsks currently being executed.
Task History	completed tasks.

3. Select the task that you want to work with.
4. Click the task step that you want to work with.
5. Click **Actions**.
6. In **Task Step Actions**, click **Location**.
The **Universal Browser** appears.
7. Select the new equipment location you want to assign this task step to.
8. If an electronic signature is required, enter the appropriate information to authorize the action.
9. Click **OK**.

Refresh

You can refresh the Task List to view all of your running tasks.

Task/Task Step Expiration

When a task or task step has expired, you can enter a statement explaining the reason for the expiration.



Note:

When using the Task List and Web Task List, you can access comments of finished workflows from the **Task History** tab. Here, users can enter expiry comments on completed or expired tasks and task steps.

Expiration Comments in the Task List

Action	Value	Description
Comment	User-defined alphanumeric sstring (A-Z, a-z, 0-9) up to 255 characters	After a task or task step has expired, a comment can be entered. <div data-bbox="1024 884 1421 1159" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: Comments may be entered multiple times; however, each entry overwrites the last one. </div>

Priority

You can apply a numeric value for a task. This priority setting does not affect the execution of the workflow; it is used only for sorting and filtering within the Task List.

Operator Environment in the Task List

Action	Value	Description
Priority	User-defined	A valid, non-negative number, such as 0, 1, 2, 3, and so on, can be entered. <div data-bbox="1024 1717 1421 1869" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: The largest number you can enter is </div>

Action	Value	Description
		 2,147,483,647 (without commas or spaces).

Related information

[Set a priority for a task or task step \(on page 2039\)](#)

Cancel

Operators can cancel an entire task or an individual task step in the Task List.

Operator Environment in the Task List

Property	Value	Description
Cancel Reason	User-defined	Required. An explanation for cancelling a task or task step can be entered.



Note:

If the task is cancelled, the continuous workflow restarts and a new task will be created.

Restrictions for Cancelling Tasks

A task must be running in order for an operator with sufficient security permissions to cancel it.

Restrictions for Cancelling Task Steps

A task step must be running or waiting to start in order for an operator with sufficient security permissions to cancel it.

Tasks Assigned to Other Operators

Anyone who has sufficient security permissions can cancel tasks or task steps assigned to other operators.

Acquire and Release

You can acquire a task step if it has not already been acquired by another operator, and then you can release it.

Logging Out with Tasks Remaining

You will be notified if you try to log out, but still have tasks to perform. You can choose to:

- release all of your task steps;
- cancel your logout, and return to your Task List to complete the tasks; or
- keep all of your task steps so that they remain acquired by you after you log out.

Restrictions on Acquiring Task Steps

You can acquire a task step only if it has not already been acquired by other operators.



Tip:

A task step may be acquired automatically if it has been preconfigured to perform this action.

Restrictions on Releasing Task Steps

You can release a task step only if you have personally acquired it.

Jump

You can skip to, or jump to, a particular task in the Task List.

How *Jump* Works

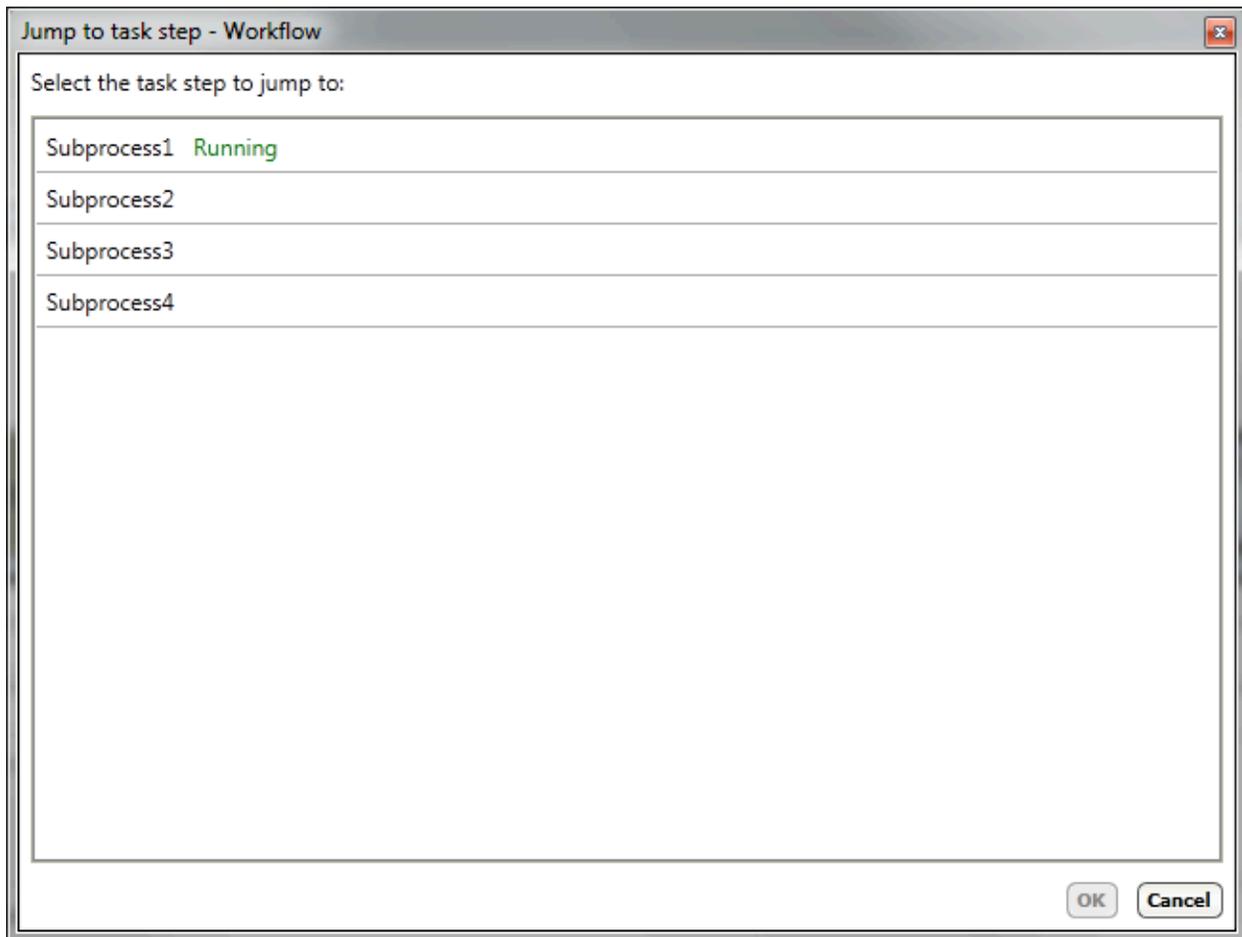
Cancelled Task Steps

Using this feature, an operator can jump back to the beginning of a task or set of task steps to do a completed or cancelled task step again. A task then resumes execution from the jump point. For example, if step 1 and step 2 have completed, and step 3 is running, and you then jump back to step 1, then steps 1, 2, and 3 execute.

Skipped Task State

If a task step is selected in the later portion of a task, all uncompleted steps previous to the selected step are skipped and their status changes to *Skipped*.

User Interface



Reassign

You can reassign a task step to a different person, piece of equipment, or location, as required. This action is performed with an active, running workflow, and it affects only the selected task step.

Task List Filter Actions

Task List Filtering

Users can use filtering to find tasks and/or task steps specific to certain criteria.

You can use the **Available Filters** panel to apply various types of filter settings. Each time you select a check box, a filter is applied, reducing your Task List further.

**Note:**

Use the filter toggle switch, , you can choose to show all tasks in your Task List, or to show tasks that match all currently applied filters. The status of your task filters, such as whether filters are applied and how many, is displayed at the top of the Task List.

- **Predefined Filters** are configured by an administrator, who sets the related criteria and values. Task List users can either enable or disable a predefined filter, but cannot modify its settings. For instructions, see *Apply a predefined Task List filter*.
- The **Quick Filter** displays the filter criteria that you can set during your current session in the Task List. If you are working in the **Filter Criteria** panel and apply filters there, the **Quick Filter** is automatically set for you. You can filter the Task List by:
 - task name
 - priority
 - step state
 - equipment and/or personnel assignment
 - expiry values
 - individual steps
- **Personal Filters** allow you to create and save specific filtering settings. These filtering settings can be viewed and used only by the user who creates them. For instructions, see *Create, save, and use a personal filter*.

You can filter the Task List by preconfigured categories. These categories organize concepts or features into groups. When you apply category filters, you focus information within your Task List.

**Note:**

Your filter selections and **Quick Filter** criteria are saved when you log out, and are re-applied when you log in again.

Related information

[Create, save, and use a personal filter \(on page 2061\)](#)

[Apply a predefined Task List filter \(on page 2060\)](#)

Filter by task name

Filter by text that is contained in the name of a task as well as searches using a wildcard character (*).

Before you begin

Depending on your access permissions and client configuration, you can select either My Task List or Task Management view

**Tip:**

Using the filter toggle switch, , you can choose to show all tasks in your Task List, or to show tasks that match all currently applied filters. The status of your task filters, such as whether filters are applied and how many, is displayed at the top of the Task List

Procedure

1. Log on to the Task List.
2. Click **Filters**.
3. To view or set a temporary filter, perform one of the following actions:
 - In the **Filter Criteria** panel, select an option.
 - Click the **Available Filters** panel, select the **Quick Filter** check box to display the **Filter Criteria** panel, and then select an option.

If you want to apply a temporary filter but have already selected one or more items under **Predefined Filters**, you must select the **Quick Filter** check box to set your filters.

**Tip:**

You can also *create and save* your own personal filters for future use.

4. Select the **Task Name** check box.
5. In the **Task Name** field, enter text related to the name of an existing task.

**Tip:**

You can use a wildcard search to match a naming pattern, such as `*taskname, taskname*`, `*taskname*`. The search is not case sensitive; however, spaces before, after, or in the middle of words are recognized.

6. To accept new or modified filtering options, click **Apply** or **Save**.

**Note:**

To exit without applying any filtering, click the **Filters** button on the control bar, and then click **Yes** in the **Filters** dialog box.

7. To remove filter(s), perform one of the following actions:

To remove...	Click...
All Filters	Clear All.
Individual filters	the associated check box.

8. Click **OK**.

Related information

[Create, save, and use a personal filter \(on page 2061\)](#)

Apply a priority filter

Filter by the number that was assigned to the task when configuring the workflow, to indicate the task's priority for filtering purposes. The number(s) specified must be greater than zero. You can filter by multiple priority numbers by separating them with a comma, such as 1,2,3. You can also filter by a range of priority numbers, such as 1-5.

Before you begin

Depending on your access permissions and client configuration, you can select either My Task List or Task Management view.



Tip:

Using the filter toggle switch, , you can choose to show all tasks in your Task List, or to show tasks that match all currently applied filters. The status of your task filters, such as whether filters are applied and how many, is displayed at the top of the Task List

Procedure

1. Log on to the Task List.
2. Click **Filters**.
3. To view or set a temporary filter, perform one of the following actions:
 - In the **Filter Criteria** panel, select an option.
 - Click the **Available Filters** panel, select the **Quick Filter** check box to display the **Filter Criteria** panel, and then select an option.

If you want to apply a temporary filter but have already selected one or more items under **Predefined Filters**, you must select the **Quick Filter** check box to set your filters.

**Tip:**

You can also *create and save* your own personal filters for future use.

4. Select the **Priority** check box.
5. In the **Priority** field, enter a number (for example, 6), a range of numbers (for example, 1-5), or a series of ranges or numbers (for example, 1-5, 10, 12-22) that represents the number that was assigned to the task when configuring the workflow, to indicate the task's priority for filtering purposes.

**Note:**

You can enter whole numbers only, such as 0, 1, 2, 3, and so on.

6. To accept new or modified filtering options, click **Apply** or **Save**.

**Note:**

To exit without applying any filtering, click the **Filters** button on the control bar, and then click **Yes** in the **Filters** dialog box.

7. To remove filter(s), perform one of the following actions:

To remove...	Click...
All filters	Clear All.
Individual filters	the associated check box.

8. Click **OK**.

Related information

[Create, save, and use a personal filter \(on page 2061\)](#)

Filter task step states

Filter by the current state of an operator's assigned task steps.

Before you begin

Depending on your access permissions and client configuration, you can select either My Task List or Task Management view.

**Tip:**

Using the filter toggle switch, , you can choose to show all tasks in your Task List, or to show tasks that match all currently applied filters. The status of your task filters, such as whether filters are applied and how many, is displayed at the top of the Task List

Procedure

1. Log on to the Task List.
2. Click **Filters**.
3. To view or set a temporary filter, perform one of the following actions:
 - In the **Filter Criteria** panel, select an option.
 - Click the **Available Filters** panel, select the **Quick Filter** check box to display the **Filter Criteria** panel, and then select an option.

If you want to apply a temporary filter but have already selected one or more items under **Predefined Filters**, you must select the **Quick Filter** check box to set your filters.

**Tip:**

You can also *create and save* your own personal filters for future use.

4. Select the **Task Step State** check box.
5. From the list, select one or more states by which to filter task steps in your Task List.

State	Description
Not Started	Task step has not been reached in the execution cycle yet.
Waiting for Start	A manual start is required.
Running	In execution.
Completed	Completed as required.
Cancelled	Cancelled by the operator or system.
Skipped	Not executed due to a jump action or a condition branch.

6. To accept new or modified filtering options, click **Apply** or **Save**.

**Note:**

To exit without applying any filtering, click the **Filters** button on the control bar, and then click **Yes** in the **Filters** dialog box.

7. To remove filter(s), perform one of the following actions:

To remove...	Click...
All filters	Clear All.
Individual filters	the associated check box.

8. Click **OK**.

Related information

[Create, save, and use a personal filter \(on page 2061\)](#)

Filter tasks by categories

You can filter the Task List by preconfigured categories. These categories organize concepts or features into groups. When you apply category filters, you focus information within your Task List.

About this task**Tip:**

To view the filtering categories associated with a task or task step, click the **Details** button.

Procedure

1. Log on to the Task List.
2. Click **Filters**.
3. To view or set a temporary filter, perform one of the following actions:
 - In the **Filter Criteria** panel, select an option.
 - Click the **Available Filters** panel, select the **Quick Filter** check box to display the **Filter Criteria** panel, and then select an option.

If you want to apply a temporary filter but have already selected one or more items under **Predefined Filters**, you must select the **Quick Filter** check box to set your filters.



Tip:

You can also *create and save* your own personal filters for future use.

4. In the **Categories** field, select the check box, and then click **Set Categories** to select a preconfigured category.
The **Universal Browser** appears.
5. Select one or more categories, and then click **OK**.
The selected items are listed in the **Categories** section.
6. To accept new or modified filtering options, click **Apply** or **Save**.



Note:

To exit without applying any filtering, click the **Filters** button on the control bar, and then click **Yes** in the **Filters** dialog box.

7. **Optional:** To remove a category, click **X**.

Related information

[Create, save, and use a personal filter \(on page 2061\)](#)

Filter operator task steps

Filter to show the current operator's assigned and active task steps only. If no active task steps are assigned to the current operator, the corresponding task is hidden.

Before you begin

Depending on your access permissions and client configuration, you can select either My Task List or Task Management view.



Tip:

Using the filter toggle switch, , you can choose to show all tasks in your Task List, or to show tasks that match all currently applied filters. The status of your task filters, such as whether filters are applied and how many, is displayed at the top of the Task List.

Procedure

1. Log on to the Task List.
2. Click **Filters**.

3. To view or set a temporary filter, perform one of the following actions:

- In the **Filter Criteria** panel, select an option.
- Click the **Available Filters** panel, select the **Quick Filter** check box to display the **Filter Criteria** panel, and then select an option.

If you want to apply a temporary filter but have already selected one or more items under **Predefined Filters**, you must select the **Quick Filter** check box to set your filters.



Tip:

You can also *create and save* your own personal filters for future use.

4. To view steps that apply to you only, select the **Only show my task steps** check box.
5. To accept new or modified filtering options, click **Apply** or **Save**.



Note:

To exit without applying any filtering, click the **Filters** button on the control bar, and then click **Yes** in the **Filters** dialog box.

6. To remove filter(s), perform one of the following actions:

To remove...	Click...
All filters	Clear All.
Individual filters	the associated check box.

Related information

[Create, save, and use a personal filter \(on page 2061\)](#)

Filter by equipment or personnel assignments

Filters task steps by the equipment and/or personnel assigned to the task step.

Before you begin

Depending on your access permissions and client configuration, you can select either My Task List or Task Management view.



Tip:

Using the filter toggle switch, , you can choose to show all tasks in your Task List, or to show tasks that match all currently applied filters. The status of your task filters, such as whether filters are applied and how many, is displayed at the top of the Task List.

Procedure

1. Log on to the Task List.
2. If applicable, select the view you want to work in.



Note:

The default view is the **Tasks** tab.

Click...	To view...
Tasks	workflow tasks currently being executed.
Task History	completed tasks.

3. Click **Filters**.
4. To view or set a temporary filter, perform one of the following actions:
 - In the **Filter Criteria** panel, select an option.
 - Click the **Available Filters** panel, select the **Quick Filter** check box to display the **Filter Criteria** panel, and then select an option.

If you want to apply a temporary filter but have already selected one or more items under **Predefined Filters**, you must select the **Quick Filter** check box to set your filters.



Tip:

You can also *create and save* your own personal filters for future use.

5. In the **Filters** panel, select the **Assignments** check box that you want to filter by.
6. To set the filter:

Click...	To filter by...
Set Location(s)	one or more equipment assignments.
Set Personnel	personnel assignments.

7. From the **Universal Browser**, select the resource(s) that you want to filter by.

8. Click **OK**.
9. To accept new or modified filtering options, click **Apply** or **Save**.

**Note:**

To exit without applying any filtering, click the **Filters** button on the control bar, and then click **Yes** in the **Filters** dialog box.

10. To remove filter(s), perform one of the following actions:

To remove...	Click...
All filters	Clear All.
Individual filters	the associated check box.

Related information

[Create, save, and use a personal filter \(on page 2061\)](#)

Filter by expiry date and/or time

Filters tasks or task steps that expire within a defined time range (from now until a defined time), or by a specific date and time. If a task has an expiry, then this filter applies to the task. If the task step only has an expiry, then this filter applies to the task step(s).

Before you begin

Depending on your access permissions and client configuration, you can select either My Task List or Task Management view.

**Tip:**

Using the filter toggle switch, , you can choose to show all tasks in your Task List, or to show tasks that match all currently applied filters. The status of your task filters, such as whether filters are applied and how many, is displayed at the top of the Task List

Procedure

1. Log on to the Task List.
2. Click **Filters**.
3. To view or set a temporary filter, perform one of the following actions:

- In the **Filter Criteria** panel, select an option.
- Click the **Available Filters** panel, select the **Quick Filter** check box to display the **Filter Criteria** panel, and then select an option.

If you want to apply a temporary filter but have already selected one or more items under **Predefined Filters**, you must select the **Quick Filter** check box to set your filters.

**Tip:**

You can also *create and save* your own personal filters for future use.

- In the **Filters** panel, select the **Expiration** check box.
- Select **Expires In** to set how long from the current time the task or task step will expire.
- Select **Expires Before** to set the date and time the task or task step is scheduled to expire.
- Click **OK**.
- To accept new or modified filtering options, click **Apply** or **Save**.

**Note:**

To exit without applying any filtering, click the **Filters** button on the control bar, and then click **Yes** in the **Filters** dialog box.

- To remove filter(s), perform one of the following actions:

To remove...	Click...
All filters	Clear All.
Individual filters	the associated check box.

Related information

[Create, save, and use a personal filter \(on page 2061\)](#)

Apply a predefined Task List filter

Predefined Filters are configured by an administrator, who sets the related criteria and values. Task List users can either enable or disable a predefined filter, but cannot modify its settings.

Before you begin

**Note:**

You can use this feature only if you have permission to do so.

Procedure

1. Log on to the Task List.
2. Click **Filters**.
3. Click the **Available Filters** panel.
The panel opens.
4. In the **Predefined Filters** section, select the corresponding check box for the filter that you want to apply.

**Note:**

Predefined filters and their settings are read-only and cannot be changed within the Task List.

Create, save, and use a personal filter

You can create and save one or more filter settings that allow you to focus the Task List according to specific criteria that you choose, such as task assignments. These filters can be viewed and used only by the operator who creates them.

Before you begin

**Note:**

You can use this feature only if you have permission to do so.

About this task

**Tip:**

You may add or remove as many personal filters as you want.

Procedure

1. Log on to the Task List.
2. Click **Filters**.
3. Click the **Available Filters** panel.
4. Click **Add**.
A new filter is added to the **Personal Filters** section.

5. Enter a name for your filter, and then click **Save** or press `Enter`.

**Tip:**

To change the name of your filter, double-click it, and then enter a new name.

6. **Optional:** To remove a filter, select the filter name to highlight it, and then click **Remove**.
7. To apply a personal filter, select the check box next to it.

**Tip:**

You can apply several personal filters at the same time.

Task List Filters

Users can perform context sensitive filtering in the Task List, which focuses a list of tasks or task steps on specific criteria.

Filtering Options for Tasks

The following table describes each available filter type.

**Note:**

Your filter selections and **Quick Filter** criteria are saved when you log out, and are re-applied when you log in again.

Type	Value	Description
Task Name	User-defined	Filter by text that is contained in the name of a task as well as searches using a wildcard character (*).
Priority	User-defined	Filter by the number that was assigned to the task when configuring the workflow, to indicate the task's priority for filtering purposes. The number(s) specified must be greater than zero. You can filter by multiple priority numbers by separating them with a

Type	Value	Description
		<p>comma, such as 1,2,3. You can also filter by a range of priority numbers, such as 1-5. These values are separated by a hyphen. If you want to filter by a series of ranges, you can separate them by commas (for example, 1-5, 10, 12-22).</p> <div data-bbox="1024 646 1419 871" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;">  Note: You can enter whole numbers only, such as 0, 1, 2, 3, and so on. </div>
My Task Steps	N/A	<p>Filter by the current operator's assigned task steps.</p> <div data-bbox="1024 1010 1419 1234" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;">  Note: All task steps that have been acquired by another operator are hidden. </div>
Task Step State	N/A	<p>Filter by the current state of an operator's assigned task steps.</p> <p>Not Started</p> <p>Task step has not been reached in the execution cycle yet.</p> <p>Waiting for Start</p> <p>A manual start is required.</p> <p>Running</p> <p>In execution.</p> <p>Completed</p>

Type	Value	Description
		<p>Completed as required.</p> <p>Cancelled</p> <p>Cancelled by the operator or system.</p> <p>Skipped</p> <p>Not executed due to a jump action or a condition branch.</p>
Categories	N/A	Filter tasks or task steps by a predefined grouping.
Location Assignment	N/A	Filters task steps by the equipment assigned to the task.
Personnel Assignment	N/A	Filters task steps by the person assigned to the task.
Expiration	User-defined	<p>Filters tasks or task steps that expire within a defined time range (from now until a defined time), or by a specific date and time. If a task has an expiry, then this filter applies to the task. If the task step only has an expiry, then this filter applies to the task step(s).</p> <div data-bbox="1024 1493 1419 1810" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;"> <p> Note: If a task or task step expires while the filter is on, the task or task step will continue to be displayed in the Task List.</p> </div>

Start Task

Start Task

The **Start Task** button allows you to start a scheduled workflow task from the Task List that has not started, yet.

**Note:**

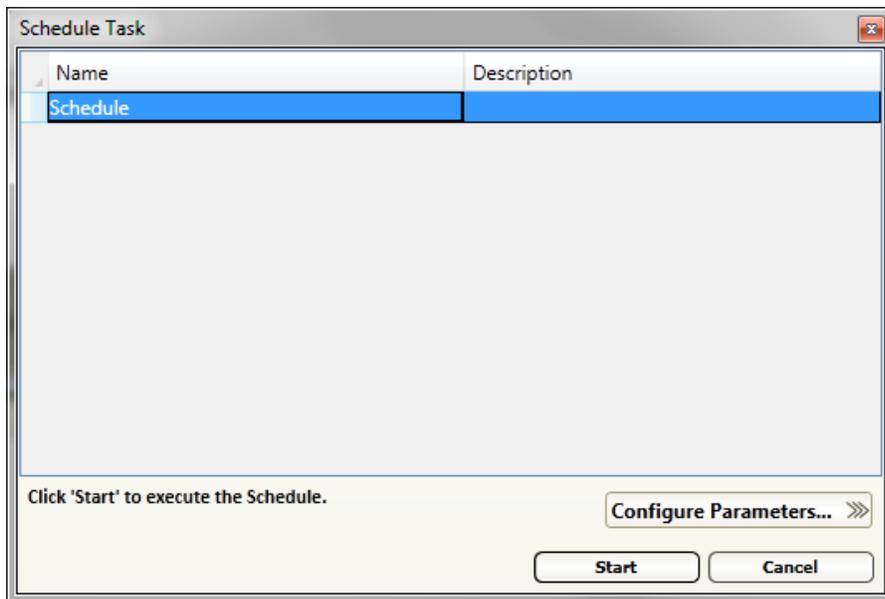
The workflow task does not wait for either a task or task step to complete; it executes immediately.

User Interface

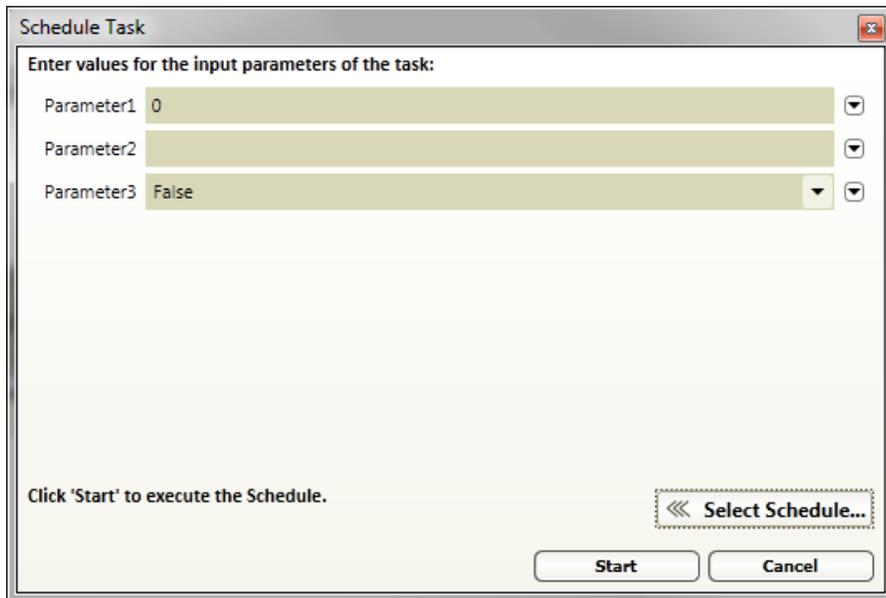
After you click the **Start Task** button to start a workflow task schedule, the **Schedule Task** dialog box appears. You can either start the scheduled task or configure parameters.

**Note:**

Starting a task may require specific operator or computer credentials. For example, only a particular operator has authority to run a task.



You can configure the input parameters for the current task execution.



Start workflow tasks manually

You can run a task in the Task List that has not started, yet.

Before you begin

Depending on your access permissions and client configuration, you can select either My Task List or Task Management view.

Procedure

1. Log on to the Task List.
2. If applicable, select the view you want to work in.



Note:

The default view is the **Tasks** tab.

Click...	To view...
Tasks	workflow tasks currently being executed.
Task History	completed tasks.

3. Click **Start Task**.
4. Select the task that you want to run.

**Note:**

If required and applicable, you can configure the workflow parameters associated with this schedule (not available on the Mobile-sized Task List and Web Task List). For more information, see *Configure parameters from Start Task*.

5. Click **Start**.

Related information

[Configure parameters from Start Task \(on page 2067\)](#)

Configure parameters from Start Task

If applicable, you can configure a task's parameters before starting that task.

Before you begin

Depending on your access permissions and client configuration, you can select either My Task List or Task Management view.

**Note:**

Parameters must have already been added to a workflow to configure them in a running workflow.

Procedure

1. Log on to the Task List.
2. If applicable, select the view you want to work in.

**Note:**

The default view is the **Tasks** tab.

Click...	To view...
Tasks	workflow tsks currently being executed.
Task History	completed tasks.

3. Click **Start Task**.
4. Select the task that you want to run.
5. Click **Configure Parameters**.
6. In a parameter field, enter a value.

7. To remove the value, click the arrow and select **Reset to Default**.



Note:

If a schedule has parameters bound to variables in other models, they are re-set to a null value and require configuration before starting the task.

8. Click **Start**.

Completed Task Information

Completed Task Information

The Task History tab displays all previously completed tasks and their task steps within a defined time range, as well as other applicable information related to these tasks.

The default time is within one hour of the current date. You can change this default by changing the date range. The time format used for history function is based on the 24-hour clock.

You can view information related to past workflows, but you cannot edit this information. Information available to view includes:

- Electronic signatures
- Input and output from all completed forms
- Personnel who:
 - completed the form
 - participated in the task
- Documents and work instructions related to the task
- Task completion date and time
- Completed forms that support reload

The Filters button can filter certain aspects of these completed tasks.

View the Task History tab

Depending on your access permissions and client configuration, you can select either My Task List or Task Management view.

Procedure

1. Log on to the Task List.
2. Click **Task History**.

View completed tasks within a date range

Depending on your access permissions and client configuration, you can select either My Task List or Task Management view.

Before you begin

**Note:**

The default timeframe is within one hour of the current search.

Procedure

1. Log on to the Task List.
2. Click **Task History**.
3. In the first Date Range field, select the start date and time, and then click **OK**.
4. In the second Date Range field, select the end date and time, and then click **OK**.
5. Click **Refresh**.

All tasks within this timeframe appear in the task grid.

Electronic Signatures in the Task List

Electronic Signatures in the Task List

Electronic signatures are collected when a task/task step action or form appears in the Task List, and an operator signs and/or verifies these activities.

When a task/task step action or form is configured to collect electronic signatures, the signatories must be the appropriate users. Only a user who is a member of the assigned group is able to sign.

**Tip:**

The font size buttons on the Task List control bar have no effect in the electronic signature section.

The electronic signature related to a completed task form can be viewed from the *Completed Task Information*.

Related reference

[Completed Task Information \(on page 2068\)](#)

Performer Signature

Electronic signatures are collected at run time when a task/task step action or form appears in the Task List, and an operator adds a signature.

When a task/task step action or form is configured to collect electronic signatures, the signatories must be the appropriate users. Only a user who is a member of the assigned group is able to sign.

Operator Entry Fields

Attribute	Value	Description
User Name	User-defined	Specifies the user name of the operator who performed the action related to task/task step or form.
Password	User-defined	Specifies the password, using masked characters, of the operator who performed the action related to task/task step or form.
Comment	User-defined	Optional. Specifies a comment entered by the operator who performed the action related to task/task step or form.
Sign	N/A	<p>Collects the signature and advances the current activity.</p> <div data-bbox="1024 1352 1422 1803" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; background-color: #E6F2FF;"> <p> Note: If a verifying signature is <u>required</u>, the Verifier Signature section appears. If a verifying signature is <u>not</u> required, the current activity advances to the next activity in the Task List.</p> </div>

Attribute	Value	Description
Cancel	N/A	<p>Stops the collection of the signature and returns the focus to the task/task step action or form.</p> <div data-bbox="1024 430 1424 787" style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px;"> <p> Note: In forms, if you cancel electronic signature collection to make changes to the form data, you must restart the signing process.</p> </div>

Information-only Fields

The status bar provides you with information about the status of your action, such as In Progress, Pending, or Completed.

Attribute	Value	Description
Authentication Type	Predefined	For security credentials, select the authentication option used for logging into Workflow; that is, Workflow or Windows.
Performer Signature	Predefined	Specifies the type of user who is entering an electronic signature.
Description	Predefined	Specifies information related to the current signing process.
Signing Group	Predefined	Displays the group that the user must be a member of in order to sign the form; for example, Administrators, Windows Users, and so on.

Verifier Signature

Electronic signatures are collected at run time when a task/task step action or form appears in the Task List, and an operator adds a signature.

When a task/task step action or form is configured to collect electronic signatures, the signatories must be the appropriate users. Only a user who is a member of the assigned group is able to sign.

Operator Entry Fields

Attribute	Value	Description
User Name	User-defined	Specifies the user name of the operator who performed the action related to task/task step or form.
Password	User-defined	Specifies the password, using masked characters, of the operator who performed the action related to task/task step or form.
Comment	User-defined	Optional. Specifies a comment entered by the operator who performed the action related to task/task step or form.
Sign	N/A	Collects the signature and advances the current activity.
Cancel	N/A	Stops the collection of the signature and returns the focus to the task/task step action or form.



Note:
In forms, if you cancel electronic signature collection to make changes to the form data, you must restart the signing process.

Information-only Fields

The status bar provides you with information about the status of your action, such as In Progress, Pending, or Completed. Information related to the **Performer Signature**, such status, operator name, and general comments, are also displayed in this section.

Attribute	Value	Description
Authentication Type	Predefined	For security credentials, select the authentication option used for logging into Workflow; that is, Workflowor Windows.
Verifier Signature	Predefined	Specifies the type of user who is entering an electronic signature.
Description	Predefined	Specifies information related to the current signing process.
Signing Group	Predefined	Displays the group that the user must be a member of in order to sign the form; for example, Administrators, Windows Users, and so on.

Sign forms electronically

Some forms may require an electronic signature in order to submit the form in the Task List.

About this task



Note:

If a Windows user is required to electronically sign a form or a task list action, the user logged into the Workflow client must also be a Windows user. A Windows user cannot sign forms when a Workflow user is signed in to the client.

Procedure

1. In the Task List, complete the form, and then click **Submit**.
The **Electronic Signature** panel appears.
2. In the **Performer Signature** area, enter the required information to authorize the action.

For example, if the workflow author configured the Form activity to use the Operator group as valid performers, then only a user who is a member of that group can sign as a performer.

- a. Select an authentication type based on your login credentials.
 - Workflow
 - Windows
 - b. If you selected:
 - Workflow, in the respective fields, enter a user name and password.
 - Windows, in the respective fields, enter a user name, password, and domain name.
 - c. **Optional:** In the **Comment** area, enter more information related to the signature.
3. Proceed based on performer signature verification requirements.

If the performer signature...	Then do this...
requires verification	Click Sign to display the Verifier Signature area, and then go to step 6. If you do not want to continue, click Cancel .
does not require verification	Click Sign and Submit . <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: In this scenario, this action represents the last step of the electronic signature process, and you do not have to complete the remaining steps of this procedure. </div>

4. In the **Verifier Signature** area, enter the required information.
- For example, if the workflow author configured the Form activity to use the Supervisor group as valid verifiers, then only a user who is a member of that group can sign as a verifier. The verifier signature cannot be completed by the personnel resource associated with the same user account as the one for the performer signature.
- a. Select an authentication type based on your login credentials.
 - Workflow
 - Windows

- b. If you selected:
- Workflow, in the respective fields, enter a user name and password.
 - Windows, in the respective fields, enter a user name, password, and domain name.
- c. **Optional:** In the **Comment** area, enter more information related to the signature.

5. Collect the verifier's electronic signature.

For a...	Click...
task/task step action	Sign and OK
form	Sign and Submit

6. **Optional:** To make changes before submission, click **Cancel Signing** to close the electronic signature panel. When the changes are complete, you must restart the signing process.

Sign task actions electronically

You may be required to add an electronic signature to a task or task step action, such as priority, cancel, jump, and reassign, in order to authorize particular activities in the Task List.

About this task



Note:

If a Windows user is required to electronically sign a form or a task list action, the user logged into the Workflow client must also be a Windows user. A Windows user cannot sign forms when a Workflow user is signed in to the client.

Procedure

1. In the Task List, navigate to a task or task step action.
 - a. In the control bar, click the **Actions** button.
 - b. Click the button for the action that you want to perform, and then enter content for the action.
 - c. Click **Sign**.
2. In the **Performer Signature** area, enter the required information to authorize the action.

For example, if the workflow author configured the Form activity to use the Operator group as valid performers, then only a user who is a member of that group can sign as a performer.

- a. Select an authentication type based on your login credentials.
 - Workflow
 - Windows
- b. If you selected:
 - Workflow, in the respective fields, enter a user name and password.
 - Windows, in the respective fields, enter a user name, password, and domain name.
- c. **Optional:** In the **Comment** area, enter more information related to the signature.

3. Proceed based on performer signature verification requirements.

If the performer signature...	Then do this...
requires verification	Click Sign to display the Verifier Signature area, and then go to step 5. If you do not want to continue, click Cancel .
does not require verification	Click Sign and Submit . <div style="border: 1px solid #0070C0; border-radius: 10px; padding: 10px; margin-top: 10px;">  Note: In this scenario, this action represents the last step of the electronic signature process, and you do not have to complete the remaining steps of this procedure. </div>

4. In the **Verifier Signature** area, enter the required information.

For example, if the workflow author configured the Form activity to use the Supervisor group as valid verifiers, then only a user who is a member of that group can sign as a verifier. The verifier signature cannot be completed by the personnel resource associated with the same user account as the one for the performer signature.

- a. Select an authentication type based on your login credentials.
 - Workflow
 - Windows
- b. If you selected:

- Workflow, in the respective fields, enter a user name and password.
- Windows, in the respective fields, enter a user name, password, and domain name.

c. **Optional:** In the **Comment** area, enter more information related to the signature.

5. Collect the verifier's electronic signature.

For a...	Click...
task/task step action	Sign and OK
form	Sign and Submit

6. **Optional:** To make changes before submission, click **Cancel Signing** to close the electronic signature panel. When the changes are complete, you must restart the signing process.

Chapter 15. Best Practices

Installation

Installation

ADAM

If ADAM is not installed in favour of a local directory, then the Proficy Publisher service does not need to run.

Workflow Performance

Workflow Performance

This section provides more information about some features, and contains suggestions for improving the performance of your workflows



Note:

Each recommendation is rated HIGH, MEDIUM, and LOW to indicate the level of importance and impact on your workflow.

Persistence

Overview

The execution engine records internal tracking data that includes the state information associated with a running workflow instance. The execution engine persists the workflow instance state when a subprocess completes. The persisted state can be used to recover and restart a workflow instance from failure conditions, or if it was stopped. If a workflow is stopped normally, the postsubprocess, postprocess, and unload steps run before the instance stops (they will not run if the server crashes). The workflow instance can continue from the start of the current subprocess by using the restart command. When this happens, the instance restarts by running the load, preprocess, and presubprocess steps. Then it restarts at the beginning of the subprocess that was running before the workflow stopped.

You can significantly improve the performance of your workflows by implementing one or all of the following recommendations:

- Decrease Persistence Points
- Reduce Activities
- Enable Persistence
- Disable Persistence

Decrease Persistence Points

Recommendation (MEDIUM)

Decrease the number of persistence points in a workflow by reducing the number of subprocesses.

If you choose to enable persistence, use as few subprocesses as possible. Since the workflow instance state is saved after each subprocess completes, unnecessary subprocesses can negatively impact the performance of the instance and any other running workflow instances.

Reduce Activities

Recommendation (HIGH)

Reduce the amount of configuration objects in a workflow.

A persisted workflow saves the state of all workflow objects. Reducing the number of activities in a workflow decreases the time needed to save the workflow instance state.

Enable Persistence

Recommendation (LOW)

Enable persistence to recover your workflow.

Workflows are automatically configured to have persistence disabled, which provides better performance of your workflow. However, you can enable persistence to:

- save the current state of the instance to the Workflow database,
- support recovery in the event of server problems,
- stop and restart your workflow.



Note:

This save process is a slow operation, impacting workflow performance as it moves from one subprocess to the next. This process is also CPU-intensive, so it can have a negative impact on the performance of other running workflow instances.

Disable Persistence

Recommendation (MED)

Disable persistence if you do not need instances of the workflow to recover after a server restart.

Workflows are automatically configured to have persistence disabled, which provides better performance of your workflow.

Memory

Overview

Workflows consume memory at run time, and all objects that are part of that workflow contribute to that memory consumption.

You can significantly improve the performance of your workflows by implementing one or all of the following recommendations:

- Reduce Concurrent Workflow Instances
- Reduce Activities

Reduce Concurrent Workflow Instances

Recommendation (MEDIUM)

Reduce the number of concurrently running workflow instances.

Every running workflow instance consumes a significant amount of server memory, even if they are instances of the same workflow definition. For example, if you have several background workflow instances that run periodically and have similar logic, combine them into a single workflow instance.

Reduce Activities

Recommendation (MEDIUM)

Reduce the amount of configuration objects in a workflow.

Each activity in a workflow uses memory at run time, including the execution state of all properties, parameters, and local variables.

Tracking Configuration

You can track workflow, subprocess, and/or schedule data while a workflow is running.

Overview

On a workflow or schedule, the system records task data for the corresponding workflow instances. It can be used to generate reports on historical data, such as task and task step durations, expiry, and performing user/location. It also allows a user of the Task List display to view previously completed tasks.

On a subprocess, specific local variables can be tracked. These variables are tracked to relate tasks to data in other models in the reporting database.

You can significantly improve the performance of your workflows by implementing one or all of the following recommendations:

- Disable Tracking
- Track Appropriate Levels
- Select Local Variables to Track

Disable Tracking

Recommendation (MEDIUM)

Disable tracking for a workflow or schedule if you do not need task instance history for the Task List display or reporting.

If tracking is enabled, then task data is saved at certain points to the workflow database. This process impacts the execution speed of the workflow.

Track Appropriate Levels

Recommendation (MEDIUM)

Enable tracking at the appropriate level.

Tracking at both the process and subprocess levels results in task data being saved more often. Therefore, to improve performance, ensure you are tracking at the appropriate levels:

- process level if historical task step data is not required
- subprocess level to view a workflow's task history in the Task List display

Select Local Variables to Track

Recommendation (LOW)

Select a minimal set of local variable for tracking.

In a subprocess, you can include specific local variables in the tracking data to relate tasks to other data. For example, you can track a work request ID in a local variable for relating task data with a corresponding work request. As a result, the number of local variables that are tracked impacts performance.

Subprocesses

Overview

A subprocess is part of a user-defined process in a workflow, and a workflow must be constructed of one or more subprocess. Each subprocess is a self-contained entity where the majority of the logic is added to run a workflow. Subprocesses group a set of related activities in a workflow and appear as a single task step. They allow you to create a series of steps to accomplish a task, assignment, event-driven, or activated process. The contents of a subprocess consist of activities that can be placed in any order depending on the requirement.

You can also make a subprocess global, which allows you to develop and store standard and frequently used subprocesses to be reused in various workflows.



Note:

When any global resource, such as a subprocess or user activity, is used in a workflow instance, it acts the same as any non-preconfigured resource. As a result, their use does not improve workflow performance.

You can significantly improve the performance of your workflows by implementing one or all of the following recommendations:

- Reduce Activities
- Set Visible to False

Reduce Activities

Recommendation (MEDIUM)

Reduce the amount of configuration objects in a subprocess improves the subprocess' starting execution time.

When a subprocess executes, the entire subprocess, including all activities, is copied. The subprocess is copied to allow it to execute repeatedly in the case of a Jump action or continuous workflow setting.

Set Visible to False

Recommendation (LOW)

Set **Visible** to **False** for a subprocess if it does not need to appear as a task step in the Task List display.

Some workflows contain subprocesses that do not have any user interaction. These subprocesses do not contain any Form or Status Message activities, and they have set the Manual Start property to **False**. Because no interaction is required, you can hide the subprocess from the Task List display by setting the **Visible** property to **False**. Setting this property improves the subprocess' execution performance and reduces the network traffic between the workflow server and the clients.

Start Workflow and Start Schedule Activities

Overview

These activities allow a workflow or a schedule to start during the execution of another workflow.

You can significantly improve the performance of your workflows by implementing the following recommendation:

- Re-use Workflow Logic

Re-use Workflow Logic

Recommendation (HIGH)

Avoid using Start Workflow or Start Schedule as a way to re-use workflow logic.

Use a global subprocess or a global user activity to re-use workflow logic between two different workflows. There is a significant performance overhead with starting a new workflow instance that does not exist with a global subprocess or user activity.

While and For Each Activities

Overview

A ForEach activity repeats the execution of a section of the workflow several times for a number of target instances in a collection, and a While activity repeats the execution of a workflow subsection as long as a specified condition is true. The condition is checked each time the activity begins.

You can significantly improve the performance of your workflows by implementing one or all of the following recommendations:

- Reduce Activities
- Minimize Iterations
- Use the Code Activity

Reduce Activities

Recommendation (HIGH)

Reduce the amount of configuration objects in the loop.

Each time a While or For Each loop iterates, it creates a copy of all contained activities, including all of the run-time property data for the activities.

Minimize Iterations

Recommendation (HIGH)

Minimize the number of iterations done by looping activities.

Minimize the number of loop iterations to minimize the number of times that the contained activities are copied and executed.

Use the Code Activity

Recommendation (LOW)

Use a Code activity to replace loops that are calculating one or more values.

If you are using a While or ForEach loop to process a list to generate one or more values, consider replacing that loop with the equivalent Code activity. This activity performs better due to the copying behaviour of the looping activities.

Write Activity

The Write activity performs a set of writes to workflow variables. The source of each write can be an expression, a bind, or a user-defined absolute value.

Write Activity Configuration

The variable browser for the Write activity displays all variables that are applicable in a given context. For example, in a subprocess you have access to all of the workflow variables (parameters, data items, and so on), all of the subprocess variables, and all of the activity properties. To add another layer, if you are at the fault handler level, then the properties of the fault handler are also available.



Note:

Data types must match when writing to parameters and variables, or when using expressions.

Combine Writes

Recommendation (LOW)

Combine multiple write activities that are in sequence.

If you have multiple write activities that are in sequence, combine these into a single Write activity that contains multiple writes to reduce execution time and memory usage.



Note:

Because the order of writes on a single Write activity is not guaranteed, use two separate write activities if one depends on the result of a previous write.

The screenshot shows a workflow editor interface. On the left, a subprocess named 'Subprocess1' is shown with a vertical flow of three Write activities: Write1, Write2, and Write3. Write1 is highlighted with a blue box. On the right, the configuration panel for 'Write1' is displayed. It has an 'Edit' link in the top right corner. Below the title, there is a section labeled 'Writes' with an 'Add Write' button. The configuration shows a single write entry for 'Subprocess1.Field1' with a value of '1'. There are control icons (three dots and an 'X') next to the field name and a dropdown arrow next to the value.

Figure 1

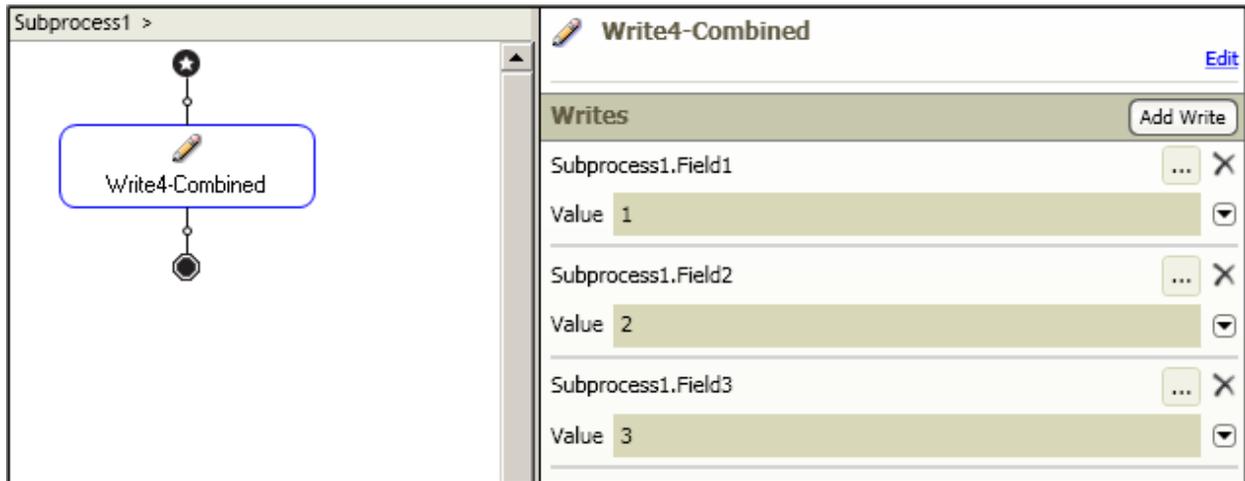


Figure 2

If/Else Activity

The If/Else activity changes execution of the workflow to one of multiple exit branches based on the resolution of a condition.

An If/Else branch can have multiple exit branches, each of which has a condition. Conditions on multiple exit flows are calculated from left to right. The flow executes on the first exit branch whose condition evaluates to True. There are no conditions on the last branch of the activity, as it is automatically set to True.



Tip:

The subprocess property value *Visible Property* can be used for this activity.

If/Else Branch

The If/Else branch contains one or more sequences as branches. Each branch has a condition that is evaluated to a boolean result. If the condition evaluates to True, the sequence for that branch is queued as the next activity for the workflow instance to execute. Conditions are evaluated from left to right, and the first sequence with a condition evaluating to True is executed. If no condition is specified, the branch serves as an Else and always executes.

Replace If/Else with Write

Recommendation (LOW)

Replace an If/Else activity with a single Write activity that uses the IF expression function.

If you are using an If/Else activity to calculate one or more values based on a series of conditions, replace it with a single Write activity. The Write activity can use the IF expression function to write different values depending on a condition, thereby reducing execution time and memory usage.

Example

If/Else activity with two branches:

- Branch 1:
 - condition of $a > 10$
 - writes a value of **valid** to a local variable
- Branch 2:
 - condition = none
 - writes a value of **invalid** to the local variable
- Value can be replaced by a single Write activity to the local variable with this expression:
 - IF($a > 10$, "valid", "invalid")

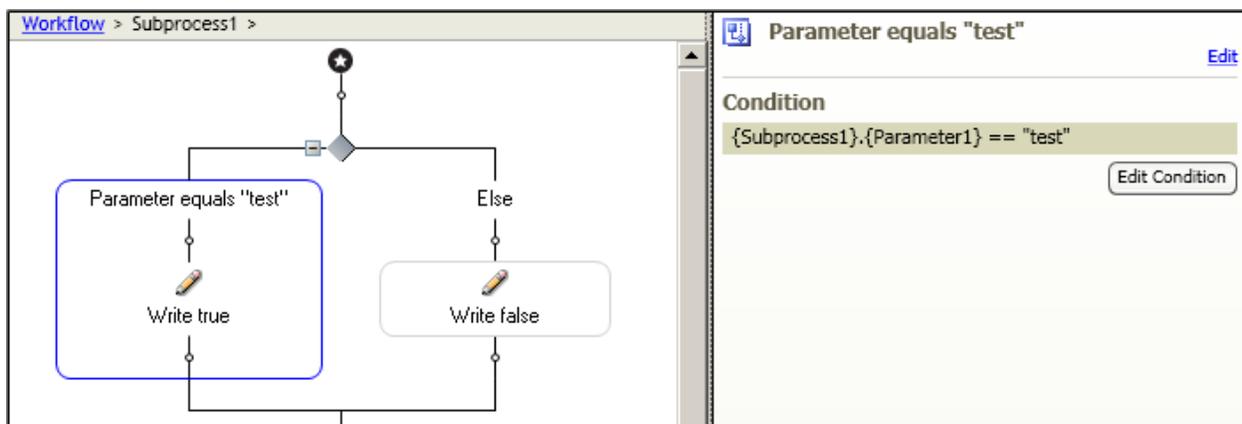


Figure 3

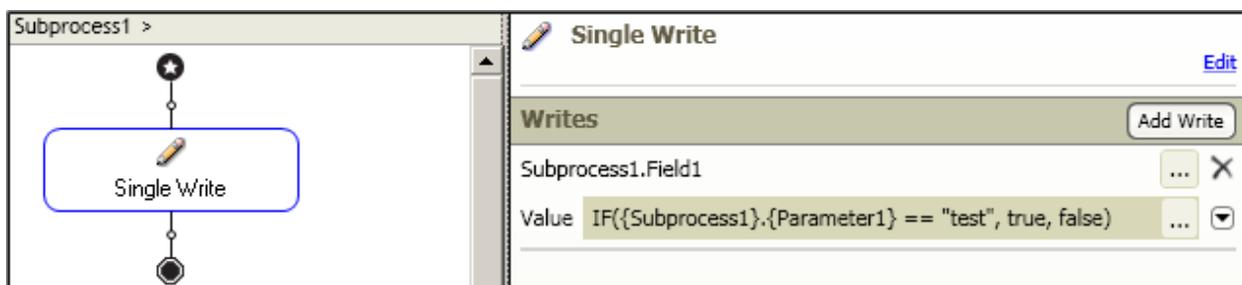


Figure 4

Form Events

Overview (HIGH)

Using Form Events to validate form data in workflow logic puts extra load on the workflow server. Though it is easier to write the validation logic in a workflow, the validation should be done in the form itself.

Forms

Forms

Overview

Forms are displays that are used as activities in workflows, providing a user interface for task steps. A form can display data or allow you to interact with a running process, or both.

You can significantly improve the performance of your workflows by implementing one or all of the following recommendations:

- Reduce Form Parameters
- Bind in the Form Control Property
- Change the Object Property
- Bind on Controls

Reduce Form Parameters

Recommendation

Minimize the number of form parameters and the size of the form parameter data.

When a form is selected for a Form activity, a property is generated for each parameter on the form. Form parameter data is transferred between the workflow server and the client Task List displays. Large form data can cause delays in the loading of the Task List display or in task change notifications sent to the display.

Bind in the Form Control Property

Recommendation

Do not bind a form control property to an output parameter value.

Binding a control property to an output parameter is not currently supported. Instead, bind the output parameter value to the control property.

Change the Object Property

Recommendation

Use the `DisplayMemberPath` property to change the object property that is displayed in collection controls.

By default, collection controls such as `ListBox` or `ComboBox` display the collection objects by using the result of the `ToString` method. This behaviour can be overridden by setting the `DisplayMemberPath` property to the name of the object property that should be displayed.

Bind on Controls

Recommendation

Do not bind to the `IsChecked.Value` property on a `RadioButton` or `CheckBox` control.

Binding to the `IsChecked.Value` property is not supported. Instead, bind directly to the `IsChecked` property.

Customize Workflow

Workflow Customization

You can customize workflows by using the existing .NET libraries.

Using Existing .NET Libraries

Overview

There are three ways that an existing .NET library can be exposed to Workflow and invoked from workflow logic.

Each approach has different advantages that may be more suitable than the others depending on the situation.

- Custom Services

In most cases, the ideal way to expose the .NET library to Workflow is to create a custom service and wrap the library. With this approach, you must create an interface for your service that contains the set of methods that workflows can invoke. In addition, you can expose events for use in triggering schedules or Wait for Event activities in workflow

- Custom Workflow Activity

Another approach for exposing a .NET library to Workflow is to create one or more custom workflow activities to wrap the library. This should only be done if the library must be invoked in the same process as the workflow engine. Some reasons for this may be if the API works with large objects that would be slow to send/receive via service methods, or if it works with objects that cannot be represented by service data contracts.

- Call from Code Activity

Custom Services

Overview

In most cases, the ideal way to expose the .NET library to Workflow is to create a custom service and wrap the library. With this approach, you must create an interface for your service that contains the set of methods that workflows can invoke. In addition, you can expose events for use in triggering schedules or Wait for Event activities in workflows.

Advantages

- Methods and events provided by a custom service are not restricted to use inside of workflows. They can be used by any application built with the application server. For example, custom service methods can be invoked by a form created with the Display Editor.
- Custom types provided by the service are published into the SOA type catalogue. This allows objects of these types to be passed between workflows and subprocesses. It also allows them to show up correctly in the workflow debugger.
- Service methods are always invoked asynchronously by workflows. This means that activities in parallel branches can continue to run while the method is invoked.

Disadvantages

- Extra effort is required to write the service wrapper over the library.
- Invoking a service method requires some additional communication overhead, especially if the service is running on a different machine. This communication may have a noticeable performance impact if large objects are passed between the workflow and the service.

Custom Workflow Activity

Overview

Another approach for exposing a .NET library to Workflow is to create one or more custom workflow activities to wrap the library. This should only be done if the library must be invoked in the same process as the workflow engine. Some reasons for this may be if the API works with large objects that would be slow to send/receive via service methods, or if it works with objects that cannot be represented by service data contracts.

For more information, see the MSDN documentation on the Windows WF 3.5 framework (the `System.Workflow` namespaces) for details on creating a custom activity.

**Note:**

Workflow does not currently support activities created with the Windows WF 4.0 framework (the `System.Activities` namespaces).

For information on registering custom activities with Workflow, see *Third-Party Activities*.

**Note:**

The Proficy Server service must be restarted to detect changes in the `ActivityExtensions.ditafile`.

Advantages

- Workflow activities are directly executed by the workflow engine and do not require the extra communication overhead of a service method call.
- Custom activities appear directly in the activity toolbox.
- Custom types that are returned by custom activities can be used by other activities by binding to properties on those objects.

Disadvantages

- Writing a set of custom activities is more complex than writing a custom service.
- The reusability of custom activities is limited because they can only be executed by workflows. They cannot, for example, be used by a form created with the display editor.
- The workflow debugger does not support custom types that are returned from activities.
- A workflow instance waits for an activity's Execute method to complete before executing other activities in the workflow instance, even in parallel branches. Therefore, care must be taken to spend a minimal amount of time in the Execute method. Note that the currently executing activity in one workflow instance will not block the execution of other workflow instances.

Call from Code Activity

Overview

If an assembly is registered with Workflow in the `ActivityExtensions.dll` file, all of its contained namespaces and types will be available to the Code activity. This means that a .NET library can be directly called by the Code activity just by registering it in the `ActivityExtensions.dll` file, even if it does not contain any custom activities.

Advantages

- This approach does not require any additional coding.
- Code activities are executed asynchronously and will not block the workflow instance from running activities in parallel branches.

Disadvantages

- If the library is exposed this way, it can only be used by workflow; it cannot be used by the Display Editor or other applications that leverage the application server.
- Custom types that are returned by code activities can only use parameters of type `Object`. This means that other activities cannot bind to the properties of those returned objects. As such, all processing of these custom types is limited to code activities.
- The workflow debugger does not debug the contents of code activities.