



# iFIX 6.1

## Using Electronic Signatures



**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.

© 2020, 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](mailto:doc@ge.com)

# Table of Contents

<b>Using Electronic Signatures</b> .....	<b>1</b>
Reference Documents .....	1
<b>Introduction</b> .....	<b>2</b>
What are Electronic Signatures? .....	2
What Determines a Signed Action? .....	3
Tracking Electronic Signatures .....	3
Features Included with Electronic Signatures .....	4
Understanding 21 CFR Part 11 .....	4
Using 21 CFR Part 11 Services from GE .....	5
<b>Getting Started</b> .....	<b>5</b>
Overview: Implementing Electronic Signatures .....	5
To implement the Electronic Signature option, you must: .....	6
License and Key Checking .....	6
Configuring Security .....	6
Using Windows User Accounts .....	7
Password Expiration Considerations .....	7
Account Lockout .....	7
Additional Considerations .....	7
Disabling the Ability to Change the System Time .....	8
Enabling Auditing in the Windows Security System .....	8
Tracking Unsuccessful Attempts to Access iFIX .....	8
Using the Security Synchronizer .....	8
Restricting Access from Remote Nodes .....	8
Understanding How Security Affects Signing .....	8
Using the iFIX Windows Screen Saver .....	10
Using Multiple Versions of iFIX in a Network .....	10
<b>Configuring Electronic Signatures</b> .....	<b>10</b>
Configuring Tags to Use Electronic Signatures .....	11
Configuring a Tag to Require an Electronic Signature .....	11

Signature Type .....	11
Signature Options .....	12
Allow Continuous Use .....	13
Perform By Comment Required .....	14
Configuration .....	14
Run Time .....	14
Exempt Alarm Acknowledgement .....	14
Understanding Unsigned Writes .....	15
Accepting or Rejecting Unsigned Writes .....	15
Implications of Database Writes With Electronic Signature .....	16
Tag Configured to Reject Unsigned Writes .....	16
Tag Configured to Accept and Log Unsigned Writes .....	16
Tag Configured to Accept Unsigned Writes .....	16
Configuring Database Manager Spreadsheet Columns .....	16
Using Operator Limits With AO Tags .....	17
To define limits directly in the picture using the Data Entry Expert: .....	17
Disabling Acknowledge All on the Alarm Summary Object .....	17
Using Comment Tables .....	18
To create a comment table: .....	18
To customize the content of a comment table: .....	19
To rename a comment table to give it a more descriptive name: .....	19
To delete a comment table: .....	19
<b>Using Electronic Signatures at Run Time .....</b>	<b>20</b>
Examining the Electronic Signature Dialog Box .....	20
How the Electronic Signature Dialog Box Works .....	24
How Do Data Links and Text Objects Behave at Run Time? .....	24
How Do Data Entry Objects Behave at Run Time? .....	24
Electronic Signature Examples .....	24
Example 1: Perform Only Signature .....	25
Example 2: Perform and Verify Signature .....	26
Example 3: Value Changes During Signing .....	27

Example 4: Selecting and Entering Comments When Signing .....	28
Example 5: Account is Disabled .....	29
Using the Alarm Summary Object at Run Time .....	30
Acknowledging a Single Alarm .....	30
Acknowledging a Page of Alarms .....	31
<b>Creating an Electronic Signature Audit Trail .....</b>	<b>33</b>
Configuring the Alarm ODBC Service .....	33
Defining the Relational Database Columns .....	34
Sending Signed Messages to a Relational Database .....	34
Example 1: Electronic Signature Not Required .....	35
Example 2: Perform Only Signature Required .....	35
Example 3: Perform and Verify Signature Required .....	36
Sending Signed Messages to Other Alarm Destinations .....	36
Example 1: Electronic Signature Not Required .....	36
Example 2: Perform Only Signature Required .....	37
Example 3: Perform and Verify Signature Required .....	37
Using the Alarm Startup Queue Service .....	37
<b>Using Electronic Signature in Scripts .....</b>	<b>37</b>
Using the Electronic Signature Object .....	38
Using Global Subroutines in Scripts for Electronic Signature .....	38
Global Subroutines and Methods that Support Electronic Signature .....	39
Global Subroutines Supported by Electronic Signature .....	39
Alarm Summary Object Methods That Support Electronic Signature .....	39
Examples: Using the Electronic Signature Object to Acknowledge Alarms .....	40
Acknowledging a List of Alarms .....	40
Acknowledging a Single Alarm .....	40
Example: Using the Electronic Signature Object to Perform a Recipe Download .....	41
<b>Testing and Troubleshooting Electronic Signatures .....</b>	<b>43</b>
Disabling Signature for Testing Purposes .....	43
Message Sent to Alarm System .....	43
Using a Local Computer with the Guest Account Enabled .....	43

Changing Tag Values in the Database Manager Spreadsheet .....	44
Determining if the Node is Enabled for Signing .....	44
Determining if a Signature is Required .....	45
Changing the Name of a Node .....	46
<b>Index .....</b>	<b>47</b>

# Using Electronic Signatures

Using Electronic Signatures is intended for application developers, process control engineers, and iFIX users who want to incorporate electronic signatures and secure electronic records (security audit trails) into their operations. This manual provides application developers with suggestions for integrating the electronic signature feature in iFIX and instructs engineers and operators on how to sign for actions.

This manual also 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.

This manual assumes familiarity with iFIX software.

## Reference Documents

For related information about subjects discussed in this manual, refer to the following manuals:

- [Setting up the Environment](#)
- [Configuring Security Features](#)
- [Implementing Alarms and Messages](#)
- [Writing Scripts](#)

## Introduction

The Electronic Signature option enables you to create a more secure environment by requiring that operators electronically "sign" for all database process changes resulting from data entry and alarm acknowledgement. Electronic signatures uniquely identify 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.

More detailed permanent records of operator actions are now written to and stored in a relational 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. The electronic signature audit trail provides greater versatility than paper trails. You can query and analyze data quickly and conveniently. Additionally, record tracking through electronic signatures increases security for process changes and alarm acknowledgements.

You can easily upgrade existing applications to take advantage of this functionality. None of your pictures need to be modified. A simple change to the tags in the Database Manager allows you to implement the Electronic Signature option. For more information, refer to [Configuring Electronic Signatures](#).

Electronic signature capability also helps address the needs of iFIX users who must conform to the 21 CFR Part 11 United States FDA government regulation. Using the feature by itself does not ensure compliance; however, applications built using the Electronic Signature option can help provide the necessary electronic verification needed to satisfy the requirements of this regulation. See [Understanding 21 CFR Part 11](#) for more information.

## What are Electronic Signatures?

Electronic signatures are the computer-generated, legally-binding equivalents of handwritten signatures. They uniquely identify the person(s) responsible for an action.

An electronic record is generated each time an action is signed for. Electronic records consist of the name of the person(s) involved in the signing process, and other details, such as the type of action performed. Electronic records are written to a relational database, and retained as a permanent record of a signed action. Refer to [Tracking Electronic Signatures](#) for more information on electronic records.

Depending on the way your tags 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 iFIX.

An electronic signature is either a Perform Only signature or a Perform and Verify signature:

**Perform Only Signature** - the operator (the "performer") that initiated the action must electronically sign for that action.

**Perform and Verify Signature** - the operator (the "performer") that initiated the action must electronically sign for that action *and* another individual (the "verifier") must electronically sign to validate the action. The action is not initiated 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 performs an action or verifies an action, a dialog box appears in which the operator must enter these two identifiers:

**User Name** - 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 iFIX user account was established using Windows security, his iFIX user name and password are the same as his Windows user name and password.

When an operator performs or verifies an action, he can optionally enter a comment related to that action. The operator can select or change a pre-defined comment, or enter an original one.

For more information on using comments with electronic signatures, refer to [Using Comment Tables](#).

## What Determines a Signed Action?

Operators can perform many actions in the iFIX run-time environment. When electronic signing is enabled in iFIX, operators may be required to electronically sign when they:

- Enter data that causes a change to the process.
- Acknowledge an alarm or a page of alarms.
- Manually delete an alarm or a page of alarms.

The need to sign for an action is determined by the way the associated tag is configured in the process database. When the application developer creates a tag, he can optionally require an electronic signature for the tag.

The application developer can configure a tag to be used by the following objects:

- Data links and text objects.
- Data entry forms, such as:
  - Slider
  - Ramp
- Alarm Summary object.
- Scripts that call global subroutines, such as:
  - WriteValue
  - ToggleDigitalPoint
  - AcknowledgeAnAlarm

When these objects are used at run time to access a tag that requires electronic signature, the operator is prompted to enter the appropriate electronic signature.

## Tracking Electronic Signatures

Each time an operator signs for 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 operators and supervisors involved in signing and verifying actions.
- Include all comments entered by the operators and supervisors involved in signing and verifying actions.
- Are recorded in a relational database via the Alarm ODBC Service.

Refer to the [Creating an Electronic Signature Audit Trail](#) chapter for complete information about configuring audit trail messages for electronic signature and examples of signed messages sent to a relational database.

## Features Included with Electronic Signatures

Using electronic signatures, you realize these benefits in your daily operations, including:

- Requiring electronic signatures for data entry and alarm acknowledgement. Refer to [Using Electronic Signatures at Run Time](#).
- Allowing operators and supervisors to add optional comments with the signature. Refer to [Using Comment Tables](#).
- Sending signed operator messages to a relational database to track operator actions. Refer to [Creating an Electronic Signature Audit Trail](#).
- Improved Windows password management, including the ability to change an expired Windows password when logging in or signing. Refer to [Configuring Security](#).
- Automatically logging out a user from an inactive workstation with the iFIX Screen Saver. Refer to the [The iFIX Screen Saver](#) section of the Configuring Security Features manual.

## Understanding 21 CFR Part 11

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 iFIX allows you to design an application that assists you in 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.

## Using 21 CFR Part 11 Services from GE

GE offers 21 CFR Part 11 consulting services to assist you with your goal of achieving 21 CFR Part 11 compliance. 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 Technical Support representative.

## Getting Started

This chapter presents an overview of the tasks required to implement the Electronic Signature option. It contains a brief description of how the option is licensed from GE. Most importantly, this chapter contains information and suggested strategies on implementing the security necessary to use the Electronic Signature option. It includes the following sections:

- [Overview: Implementing Electronic Signatures](#)
- [License and Key Checking](#)
- [Configuring Security](#)
- [Using Windows User Accounts](#)
- [Understanding How Security Affects Signing](#)
- [Using the iFIX Windows Screen Saver](#)
- [Using Multiple Versions of iFIX in a Network](#)

## Overview: Implementing Electronic Signatures

The steps that follow provide an overview of the steps to implement electronic signatures into your iFIX application.

► **To implement the Electronic Signature option, you must:**

1. Ensure that both the computer to be used to enter electronic signatures and the computer to be used as the SCADA node are equipped with hardware keys that have the Electronic Signature option enabled.
2. Establish the appropriate security configuration, which includes:
  - a. Enabling security and creating users and groups. This may be done using the iFIX Security Configurator, or by using the iFIX Security Synchronizer. Refer to [Configuring Security](#) and [Using the Security Synchronizer](#) for more information.
  - b. Assigning the appropriate security areas to those users and groups.
  - c. Assigning the appropriate application features to those users and groups. The application features available for electronic signature are:
    - **Electronic Signature - Perform By** - Grants the user the ability to perform and sign for actions.
    - **Electronic Signature - Verify By** - Grants the user the ability to verify actions that another user performs.
3. Configure tags to require electronic signature.
4. Configure the Alarm ODBC Service and your relational database. You must perform this step if you want to provide an audit trail of your process.

## License and Key Checking

To use the Electronic Signature option, you must purchase the option from GE and receive hardware keys with this functionality enabled. You must install the keys on both the SCADA node and on the iClient node. The application developer typically configures tags on the SCADA node, and operators typically enter electronic signatures on the iClient node. The keys are checked at run time, when an object whose tag requires electronic signature is selected. Both the iClient and the SCADA keys must have the Electronic Signature option enabled to use this functionality.

When the Electronic Signature option is enabled, you may be required to sign for actions you perform during run time. When the Electronic Signature option is disabled, you can perform actions without needing to sign for them.

Refer to [Determining if the Node is Enabled for Signing](#) for details on determining the status of the node.

## Configuring Security

To use the Electronic Signature option, you must first enable iFIX security. Once security is enabled, you must assign the appropriate application features to the users who will use this option. You can perform both of these tasks in the iFIX Security Configuration program. If your application uses security areas on tags, you will also need to make sure that these same users also have rights to those security areas.

If you want to build an application with the goal of achieving compliance with the 21 CFR Part 11 regulation, it is strongly encouraged that you use Windows user accounts when using the Electronic

Signature option within iFIX. Windows user accounts allow for password expiration and account lock-out, which ensures a more secure signing environment. When your Windows password expires, you can change it without leaving iFIX, either when you log in or when you enter an electronic signature.

Refer to the [Using iFIX With Windows Security](#) chapter in the Configuring Security Features manual for more information on using Windows user accounts.

## Using Windows User Accounts

It is encouraged that you use Windows user accounts to provide a more robust security environment, as either part of a strategy for 21 CFR Part 11 or as a means to provide an additional level security within any operation. By leveraging this functionality, you can add password expiration control and account lockout to your overall security environment.

For more information on using Windows security, refer to the [Using iFIX With Windows Security](#) chapter in the Configuring Security Features manual.

## Password Expiration Considerations

When a user logs in or enters an electronic signature at run time:

- 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.

If you do not want passwords to expire, you can enable the Password Never Expires option in the Windows security configuration. If you do not want operators to change passwords, you can enable the User Cannot Change Password option in the Windows security configuration.

## Account Lockout

The application developer 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 a user logs in or enters an electronic signature at run time, he receives an error if the account has been disabled. The application developer can configure the message to display with the error, such as a telephone number or the name of a contact person; otherwise, a general message displays.

Refer to the [Configuring the Account Disabled Message in iFIX](#) and [Limiting the Number of Invalid Login Attempts](#) sections in the Configuring Security Features manual for information on setting the account lockout threshold and configuring the account disabled message.

## Additional Considerations

This section contains some suggested strategies for configuring a 21 CFR Part 11 environment.

## Disabling the Ability to Change the System Time

Application developers may want to disable an operator's ability to change the system time by removing the "Change the system time" user right from the appropriate user accounts in Windows security. By doing so, you can prevent inaccurate timestamps from entering the audit trail.

## Enabling Auditing in the Windows Security System

Application developers who want to monitor Windows security events, such as logon and logoff, should enable auditing in the Windows Local Security Policy. You can display these events in the Windows Event Viewer's security log.

## Tracking Unsuccessful Attempts to Access iFIX

Each time an unsuccessful attempt is made to access the iFIX system, a message is sent to the alarm system. If you have configured the Alarm ODBC Service and your relational database, these messages are also written to your relational database, and can be included in the audit trail of your process.

Refer to the [Understanding the Security Log File](#) section in the Configuring Security Features manual for more information.

## Using the Security Synchronizer

A new tool, the Security Synchronizer, is available to help synchronize your iFIX user accounts with your Windows user accounts. The Windows-to-iFIX Security Synchronizer provides a single point of configuration for management of user accounts. This application assists customers who want to create Windows user accounts to produce a more secure environment.

The person who administers the Windows security system adds and removes users from specific Windows groups. The Security Synchronizer application creates, modifies, and deletes iFIX user accounts based on information retrieved from the Windows security system. This allows you to administer security in Windows and have those changes propagated to iFIX. When you are using the Security Synchronizer, all modifications are made to the iFIX security configuration; the Windows security configuration is not modified.

Refer to the [Using Security Synchronizer](#) section of the Configuring Security Features manual for complete information about configuring and using this tool.

## Restricting Access from Remote Nodes

Application developers can allow certain remote nodes the ability to write to specific SCADA nodes only. This prevents the possibility of access from unknown or unauthorized nodes. This is an important feature to ensure that operators are positioned physically close to the equipment they are manipulating. You may want to incorporate this feature to provide a more secure environment for your SCADA nodes.

Refer to the [Protecting SCADA Nodes](#) section in the Configuring Security Features manual for more information.

## Understanding How Security Affects Signing

The following example illustrates how security is used when a user enters an electronic signature. For this example, two security areas are created: Line 1 and Line 2. Four user accounts are established: George, Thomas, Peter, and Laura. Each user has a different job responsibility with access to various application features and security areas. This example assumes that each user account is connected to a Windows user account.

The following table shows each user's job responsibility and the iFIX application features and security areas assigned to each.

User	Job Responsibility	Application Feature(s)	Security Area(s)
George	Operator for Line 1	Electronic Signature - Perform By	Line 1
Thomas	Operator for Line 2	Electronic Signature - Perform By	Line 2
Peter	Senior Operator for Line 2	Electronic Signature - Perform By Electronic Signature - Verify By	Line 2
Laura	Supervisor	Electronic Signature - Perform By Electronic Signature - Verify By	Line 1, Line 2

The following table shows the tags that represent the set point for each line. Each tag is configured with a signature type of Perform and Verify and has been assigned the appropriate security area.

Tag Name	Security Area
SETPT1	Line 1
SETPT2	Line 2

The following table shows which users can perform and verify for actions on each line.

Security Area	Perform	Verify
Line 1	George, Laura	Laura
Line 2	Thomas, Peter, Laura	Peter, Laura

The following table shows a set of operator actions and the result at run time.

Operator Action	Result	Reason
George enters a new value for SETPT1 and signs for it. Laura verifies this action.	Action succeeds; the value is written to the database.	George and Laura have access to the Line 1 Security Area; George has access to the Perform By application feature; Laura has access to the Verify By application feature.
Thomas enters a new value for SETPT2 and signs for it. Peter verifies this action.	Action succeeds; the value is written to the database.	Thomas and Peter have access to the Line 2 Security Area; Thomas has access to the Perform By application feature; Peter has access to the Verify By application feature.
George enters a new value for SETPT2 and attempts to sign for it.	Action fails.	George does not have access to the Line 2 Security Area.
Peter enters a new value for SETPT2 and signs for it. Thomas attempts to verify it.	Action fails.	Thomas does not have access to the Verify By Application feature.

Peter enters a new value for SETPT2 and signs for it. Peter also attempts to verify it.	Action fails.	The same person cannot perform and verify an action.
George enters a new value for SETPT1 and signs for it. He enters a value of 90. The tag's EGU high limit is set to 100, and its operator high limit is set to 80.	Action succeeds. The new value 90 is written to the tag, but the tag clamps the value at 80.	George cannot exceed the operator limit of 80. Although he signed for 90, the tag accepts only up to 80.

## Using the iFIX Windows Screen Saver

As part of your strategy to secure inactive computers, you can use the iFIX Screen Saver. You can configure the iFIX Screen Saver to perform many functions, such as terminating the continuous use period or logging out the current iFIX user when the Screen Saver activates.

Refer to the [The iFIX Screen Saver](#) section in the Configuring Security Features manual for complete details on configuring the iFIX Screen Saver.

## Using Multiple Versions of iFIX in a Network

When using electronic signature, it is always best to use 4.0 SCADA and 4.0 iClient nodes. Nodes running earlier versions may not respond properly to electronic-signature enabled tags.

If you open a 4.0 database from an older-version node, such as 2.6, you cannot add or modify individual tags, but you can otherwise modify the database. For example, you can delete and duplicate tags. You cannot open an older-version node from a 4.0 node.

To enable electronic signature, your SCADA node must be a 4.0 node.

If you use multiple versions, the older version node may produce this type of message:

```
No message exists for error <error number>.
```

The error number will display as a literal, such as 1798. This message is likely caused by a signature-related error, but the older version node does not support electronic signature functions. Consequently, the message does not display properly.

If you use multiple versions of iFIX, you run the risk of acknowledging alarms configured for electronic signature without capturing a signature for them.

If your intention is to create an application for use in a 21 CFR Part 11 environment, you *must* use a 4.0 (or greater) SCADA and 4.0 (or greater) iClient node.

**IMPORTANT:** If you configure a tag to require electronic signature on a 4.0 SCADA node and then acknowledge an alarm for that tag on a 2.6 or earlier iClient node, the Electronic Signature dialog box will not appear, and the operator will be able to acknowledge the alarm without entering a signature. It is strongly recommended that you use only 4.0 SCADA and iClient nodes when using electronic signature.

## Configuring Electronic Signatures

This chapter introduces the tasks the application developer must complete to enable electronic signatures in iFIX. The topics covered in this chapter include:

- [Configuring tags to use the Electronic Signature option.](#)
- [Disabling AcknowledgeAll on the Alarm Summary object.](#)
- [Creating comment tables.](#)

## Configuring Tags to Use Electronic Signatures

You can configure each tag in the process database to require a signature, including built-in block types, such as Analog Output (AO) and Digital Output (DO), and any other Database Dynamo (also known as a loadable block) that has been updated to support electronic signature. Signature options and security areas for each tag are configured by the application developer in the Database Manager for operators to use.

If you want to use an OPC server, you can pull that data into the iFIX process database using the OPC Client Driver, and then access the data through the tag. The tag must be configured to require electronic signature.

You cannot configure system tag fields, NSD fields, and alarm counters fields for electronic signature. To help maintain a secure environment, you should avoid using these tags when creating pictures that operators will use.

**CAUTION:** Exercise caution when enabling electronic signature in existing databases. Some tags may be written to from custom programs and scripts. Exporting the database and changing all tags to require electronic signature may cause custom programs and scripts to function improperly.

## Configuring a Tag to Require an Electronic Signature

The Electronic Signature options for each tag are located on the Advanced tab of the tag's configuration dialog box, and are available when you create or modify a tag. These options are:

- [Signature Type](#)
- [Signature Option](#)
- [Unsigned Writes](#)

The sections that follow describe each option.

### Signature Type

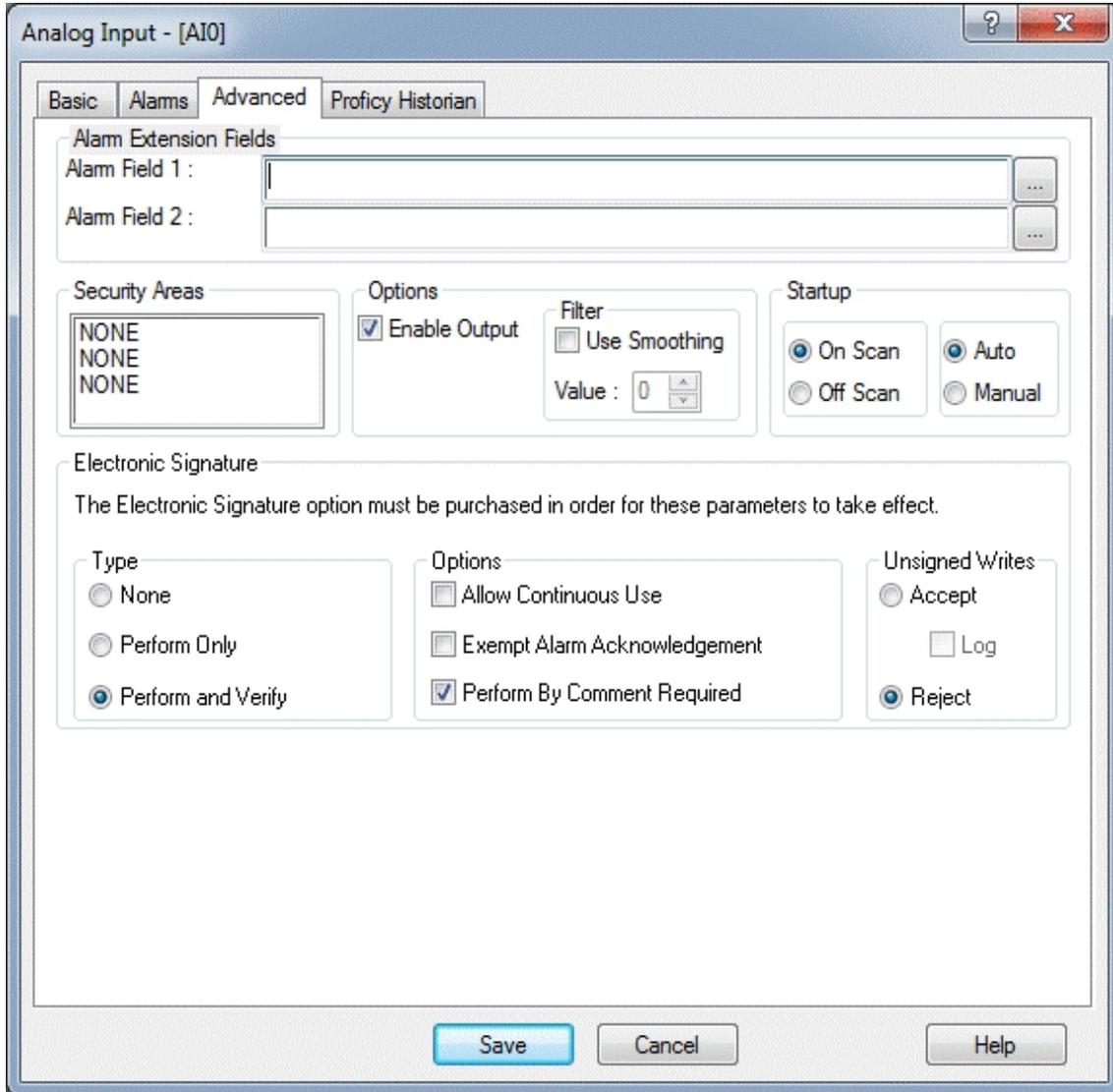
The following fields are available for the Signature Type option:

**None** - Do not require signature for this tag. This option is the default setting.

**Perform Only** - Require only the signature of the operator performing the action.

**Perform and Verify** - Require the signature of the operator performing the action *and* the signature of another user to verify the action.

For example, tag AO\_3 is configured to require a Perform Only signature, as indicated in the following figure.



*Enabling the Electronic Signature Option in a Tag's Dialog Box*

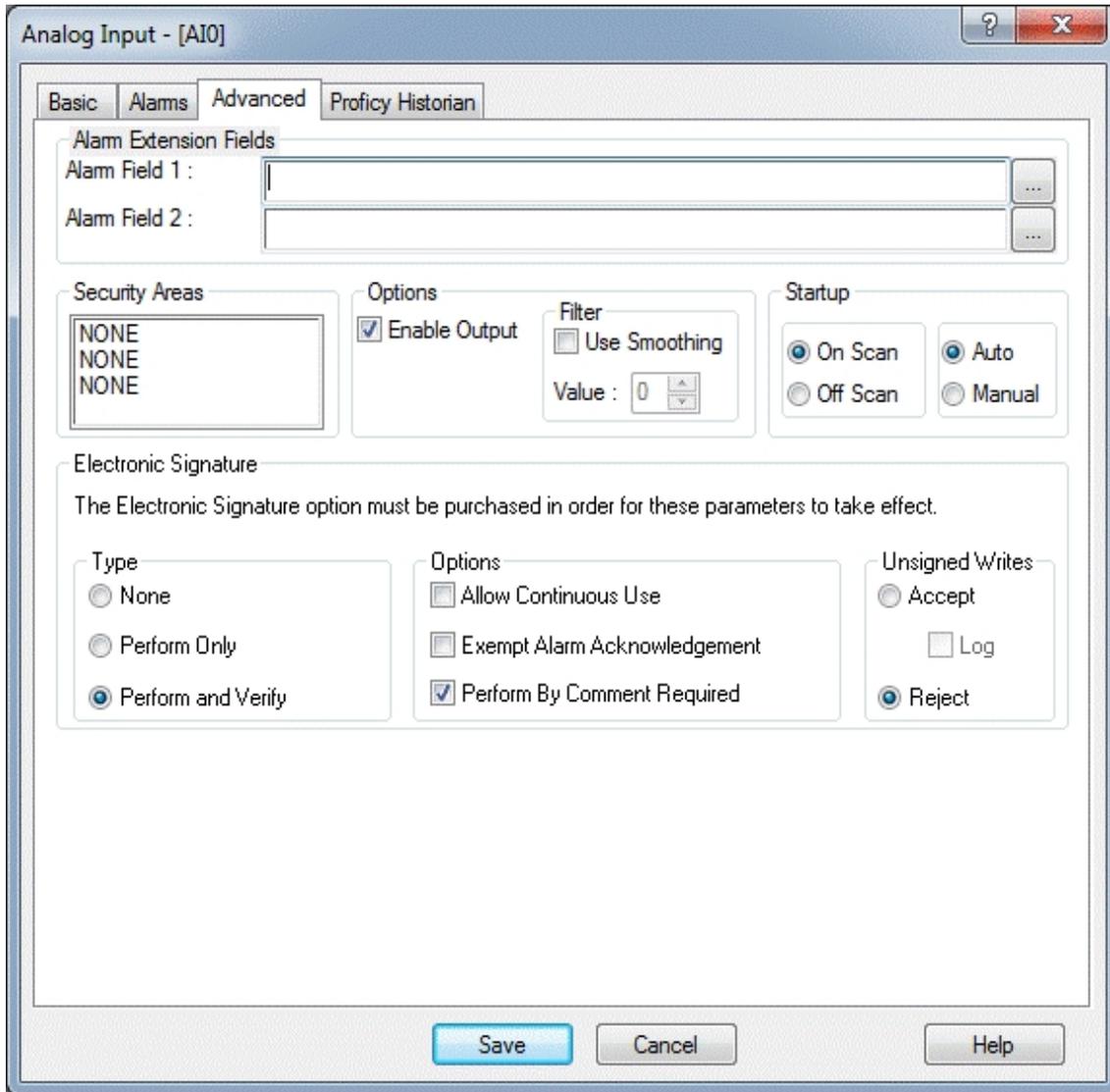
## Signature Options

The following fields are available for the Signature Option on the Analog Output.

**Allow Continuous Use** - When enabled, lets you repeatedly sign for successive actions by supplying only a password. For more information, on this option, refer to the [Allow Continuous Use](#) section.

**Exempt Alarm Acknowledgment** -When enabled, allows you to configure whether Alarm Acknowledgments and Manual Alarm Deletions will require an electronic signature.

**Perform By Comment Required** - When enabled, requires you to enter a comment to perform in run mode when the Perform Only (or Perform and Verify) type is enabled. For more information, refer to the [Perform By Comment Required](#) section.



### Allow Continuous Use

The Allow Continuous Use check box is selected in the [Enabling the Electronic Signature Option dialog box](#) figure. By default, continuous use is enabled. This option allows the operator to repeatedly sign for successive actions by supplying only a password. A continuous use period starts when the operator successfully signs for an action. The operator's user name is recorded as the continuous user. While the continuous use period is in effect, the operator's user name displays in the Performed By section in each subsequent Electronic Signature dialog box displayed for this tag. This feature saves the operator the

time required to repeatedly type the user name. Continuous use applies only to the person performing an action and does not affect the person verifying an action.

A continuous use period ends when the iFIX Screen Saver activates due to a period of keyboard or mouse inactivity, or when another user enters his user name in the Perform By section of the Electronic Signature dialog box. The application developer configures the duration of the inactive period and must also enable the Reset Electronic Signature Continuous User option for the Screen Saver.

When the continuous use period ends, the Performed By user name is cleared in the Electronic Signature dialog box. When a valid signature is entered, another continuous use period begins.

**NOTE:** The name of the continuous user changes and the continuous use period restarts each time a different user enters a valid signature.

## Perform By Comment Required

When selected, the Performed By Comment Required option enables Comment enforcement in the Perform Comment section. This means that the operator must enter comments in the Performed By Comment box in the Electronic Signature section during run mode. Comments in the Verify Comment section are optional.

### Configuration

In conjunction with the Perform By Comment Required option, you must select the Electronic Signature type as one of the following for that specific writable tag:

- Perform Only
- Perform and Verified

The Perform by Comments Required check box appears in every block configuration where Electronic Signature settings are available. By default, this check box is disabled.

You can also configure this option to display an additional column in the Database Manager tag list.

- In Ribbon view, navigate to Settings > Properties. The eSig Comment Required property appears in the Column > Available Columns section. You can use the Add button to select to display this information in your Database Manager tag list.
- In Classic view, navigate to View > Properties. Select the eSig column you want to display from the Available Column listing. Use the Add button to add it to the Display Columns, which makes this column visible in the Database Manager tag list.

### Run Time

During run time, if you enter a setpoint value for a tag, or if you acknowledge alarms, then the electronic signature dialog appears depending on the option configured for the tag. If a tag is configured with Perform By Comment Required enabled, you must enter a comment in the appropriate edit box. The comment must be more than 1 character but less than 168 characters. An error message box appears when the comment does not meet these specifications.

For more information on these dialog boxes, refer to the [Examining the Electronic Signature Dialog Box](#) section.

## Exempt Alarm Acknowledgement

The Exempt Alarm Acknowledgement check box allows you to configure whether Alarm

Acknowledgements and Manual Alarm Deletions will require an electronic signature. By default, once you configure the tag to require signature, these actions require a signature. By selecting this check box, you can choose not to have these actions require a signature. Therefore, the Electronic Signature dialog box will not display each time an operator acknowledges or manually deletes an alarm.

## Understanding Unsigned Writes

While the most common data entry objects support electronic signature, there are ways to write to a tag without capturing a signature. This is called an unsigned write. The application developer can specify whether to allow these writes to update the database (accept) or to block (reject) them on a tag-by-tag basis. Unsigned writes can originate from:

- Scripts.
- Recipe downloads.
- Use of the Acknowledge All command on the Alarm Summary object or the AcknowledgeAllAlarms global subroutine.
- Global subroutines called from Scheduler scripts.
- Applications other than the WorkSpace, such as the Scheduler, Database Manager, or externally-written EDA applications.
- Writes to data sources that use expressions.
- Writes that originate from an iFIX 2.x or FIX32 node.

### Accepting or Rejecting Unsigned Writes

You can configure a tag to accept or reject an unsigned write. In a secure signing environment, it is typical to reject unsigned writes; by default, the Reject Unsigned Writes option is selected.

The following fields are available in the Unsigned Writes section:

**Accept** - the tag accepts the write from an unsigned source.

**Log** - sends a message to the alarm system, indicating that the tag accepted an unsigned write. This field is available only when the Accept option is selected.

**Reject** - the tag rejects a write from an unsigned source in the same manner a write is rejected for a security violation. A message is sent to the alarm system to flag the violation. By default, this option is selected.

If a user changes a field of a tag that requires electronic signature directly in the Database Manager spreadsheet, that change is considered an unsigned write. Writes from the Database Manager tag configuration dialogs and SAC are always accepted, regardless of how you configure unsigned writes. Refer to the [Changing Tag Values in the Database Manager Spreadsheet](#) section in the Testing and Troubleshooting Electronic Signature chapter for details.

Values written from signature-disabled nodes to tags that require signature may be rejected. If an application has mixed nodes (some with the Electronic Signature option enabled, some with the option disabled), the application developer can restrict access to certain remote nodes to disallow writes from the disabled nodes, if necessary. For more information about disabled nodes, refer to [Restricting Access](#)

[from Remote Nodes](#). For more information about mixed nodes, refer to [Using Multiple Versions of iFIX in a Network](#).

## Implications of Database Writes With Electronic Signature

You must understand the implications of database writes in a secure environment when electronic signature is enabled. Unsigned writes do not prompt for electronic signature. However, you can trace the results of unsigned writes through the alarm system. This section discusses the behavior of unsigned writes, based on the tag's configuration. To learn how to configure a tag for unsigned writes, refer to [Understanding Unsigned Writes](#).

### Tag Configured to Reject Unsigned Writes

If the tag is configured to reject unsigned writes, the write is rejected and this type of message is written to the alarm system:

```
UNSIGNED WRITE REJECTED: <tag name> cannot be written without electronic signature by <logged-in user name>
```

### Tag Configured to Accept and Log Unsigned Writes

If the tag is configured to accept unsigned writes *and* to log the event, the write is accepted, and this type of message is written to the alarm system:

```
UNSIGNED WRITE ACCEPTED: <tag name> was written without electronic signature by <logged-in user name>
```

### Tag Configured to Accept Unsigned Writes

If the tag is configured to accept unsigned writes, but not to log the event, the write is accepted, and this type of message is written to the alarm system:

```
<tag name> set to <new value> by <node>::<logged-in user name>
```

## Configuring Database Manager Spreadsheet Columns

To view electronic signature settings for a tag, you can add these columns to the Database Manager spreadsheet using the Column tab of the Database Manager's Properties dialog box:

**eSig Type** - Indicates the signing requirements on this tag. When you add this column, the following values display:

**NONE** - The tag is not enabled for electronic signing.

**PERFONLY** - The tag requires only the signature of the operator performing the action.

**PERVERI** - The tag requires the signature of the operator performing the action and the signature of the person verifying the action.

**eSig Cont Use** - Indicates if allow continuous use is enabled for this tag.

**eSig Exempt Ack** - Indicates if signing is required for alarm acknowledgement and manual alarm deletion on this tag.

**eSig Unsigned Writes** - Indicates if unsigned writes are accepted or rejected by this tag. When you add this column, the following values can display:

**ACCEPT** - This tag is configured to accept unsigned writes.

**LOG** - If this tag is configured to accept unsigned writes, a message is sent to the relational database whenever an unsigned write is accepted.

**REJECT** - This tag is configured to reject unsigned writes. A message is sent to the relational database whenever an unsigned write is rejected.

For more information on changing database columns, refer to the [Locating and Displaying Data](#) chapter of the Building a SCADA System manual.

## Using Operator Limits With AO Tags

You can configure AO blocks to have both engineering unit limits and operator limits. You set these limits, respectively, in the Basic tab and Advanced tab of the Analog Output dialog box. The value you set for the engineering unit limit overrides the value you set for the operator limit if the operator limit is outside the range of the engineering limit. For example, you can set an AO tag to have a high engineering limit of 100, and you can set the operator limit for this tag to be 80. If an operator changes the value of this tag to 90, the value written to the tag is 90, but the tag clamps the value at 80, the operator limit.

The message sent to the audit trail reflects the value the operator signed for, in this example, 90. If the AO tag has Alarming and Event Messaging options enabled, an event message is sent by the AO tag that reflects the clamped value, in this example, 80.

You can also define limits directly on data entry objects used in pictures.

### ► To define limits directly in the picture using the Data Entry Expert:

1. Click the Data Entry Expert button to open the Data Entry Expert dialog box.
2. In the Data Source field, define your data source.
3. Clear the Fetch Limits from the Data Source check box. If you enable this field, the limits are derived from the engineering unit limits defined for the data source, and the limits you define here are overridden.
4. In the Low Limit and High Limit text boxes, enter preferred low and high limits.

**NOTE:** These limits are checked before the limits on the tag are checked.

## Disabling Acknowledge All on the Alarm Summary Object

The Electronic Signature option does not support Acknowledge All alarms capability. It is encouraged that you disable this function from the Alarm Summary object by clearing the Allow Acknowledge All Alarms check box on the Operator tab of the Alarm Summary Configuration. This removes the Acknowledge All option from the right-click menu and prevents an operator from acknowledging all alarms from the Alarm Summary object.

**IMPORTANT:** If you do not disable the Allow Acknowledge All Alarms option, you risk the chance of allowing operators to acknowledge alarms in an unsecured environment. This option is enabled by default in pictures created in iFIX 3.0 and higher.

Operators cannot sign for alarm acknowledgements when using AcknowledgeAll through the Alarm Summary object or through a script.

## Using Comment Tables

You may want operators to add specific comments to their electronic signature at run time, either when performing an action or when verifying an action. An operator may enter a comment about the condition of a machine, such as "Conveyor jammed." and the person who verifies that action may enter a comment about the resolution, such as "Problem fixed. Realigned station 14."

You can create pre-defined comments for the operator to select from when entering an electronic signature. To create a pre-defined comment, you must create comment tables. These tables can populate the Predefined Comment list of the Electronic Signature dialog box with your comments. The operator can also write a free-form comment in the Comment field.

**NOTE:** The operator can enter text in the Comment field regardless of whether you configure a comment table and create pre-defined comments.

### ► To create a comment table:

1. In the iFIX WorkSpace, in Ribbon view, on the Home tab, in the WorkSpace group, click Settings, then select User Preferences.  
- Or -  
In Classic View, on the WorkSpace menu, click User Preferences.
2. Select the General tab.
3. Optionally, rename the table(s) by entering text in the Perform Comments Table Name field and the Verify Comments Table Name fields, respectively.
4. Click the corresponding button, either the Create Perform Comments Sample Table button or the Create Verify Comments Sample Table button, as appropriate. A message box appears, confirming that the comment table was successfully created. If you do not rename the tables in step 3, these default names are assigned to the tables:
  - PerformESigComments
  - VerifyESigComments

The comment tables you create are listed in the System Tree of the WorkSpace, in the Globals/User folder, as indicated in this graphic:



5. Click OK in the WorkSpace message box to save the changes in the user.fgx file.

Once you create a comment table, you can fill it with comments you want to use in your application.

► **To customize the content of a comment table:**

1. Right-click the comment table, such as PerformESigComments, in the System Tree, and select Custom. The Custom Lookup Table dialog box appears.
2. In the Value column, enter a numeric value. This value can be up to seven digits long. Sequence the numbers in this column in relative, ascending order. This value is used as an index for this comment and is not otherwise associated with the comment.
3. In the String field, enter the comment. The comment can be up to 168 characters long. Keep in mind that this comment can be combined with a free-text comment in the Electronic Signature dialog box, so you may want to conserve the length of this comment.

► **To rename a comment table to give it a more descriptive name:**

1. Open the Properties window using a standard method, such as right-clicking a comment table in the System Tree and selecting the Property Window... option.
2. Access the System Tree.
3. Select the comment table you want to rename in the System Tree.
4. Return to the Properties window, select the Name property, and enter the new table name.
5. Close the Property window.

► **To delete a comment table:**

1. Access the System Tree.
2. Right-click the name of the table you want to delete, and click the Delete option.

## Using Electronic Signatures at Run Time

This chapter describes how Electronic Signatures are used in the run-time environment, and describes the tasks required of an operator when electronic signature is required. It includes the following sections:

- [Examining the Electronic Signature Dialog Box](#)
- [How the Electronic Signature Dialog Box Works](#)
- [Electronic Signature Examples](#)
- [Using the Alarm Summary Object at Run Time](#)
- [Acknowledging a Single Alarm](#)
- [Acknowledging a Page of Alarms](#)

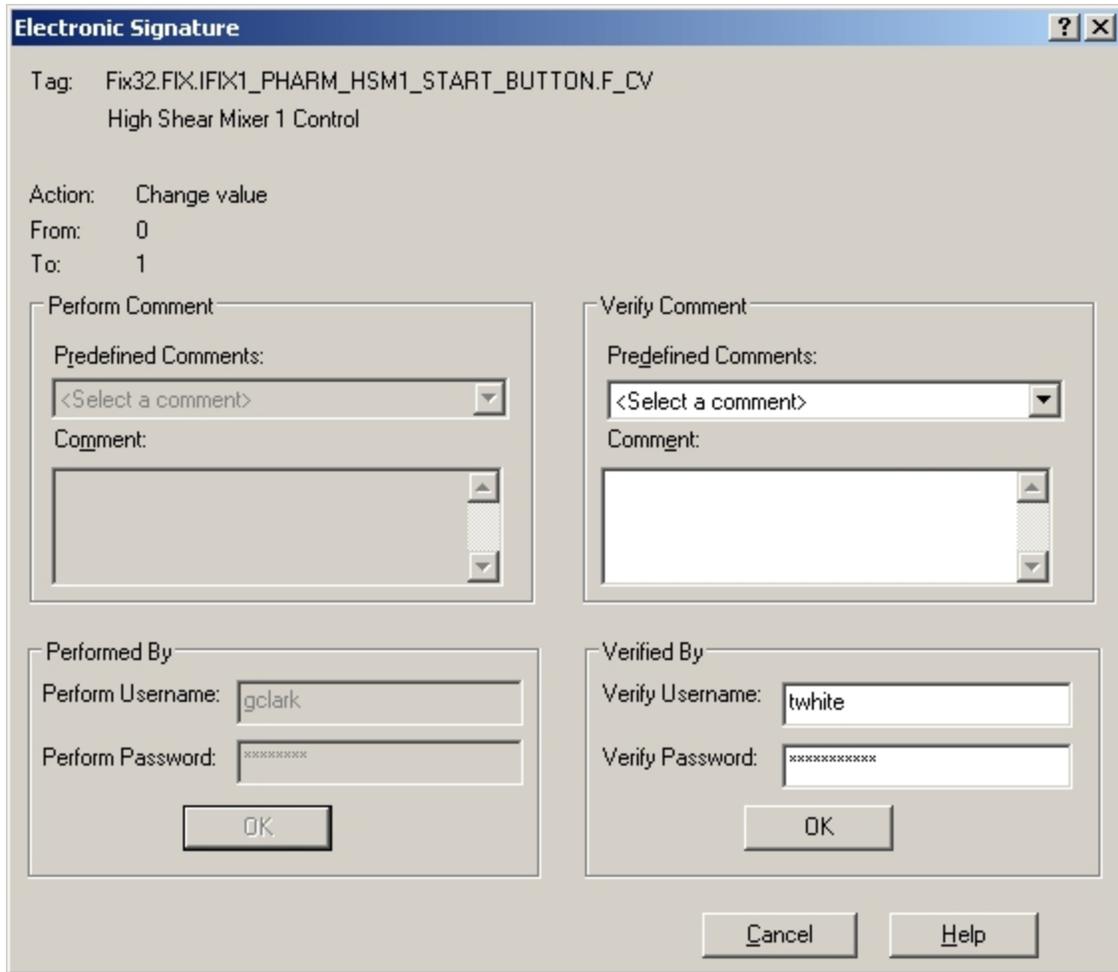
### Examining the Electronic Signature Dialog Box

Before using electronic signatures at run time, you should be familiar with the fields used in the Electronic Signature dialog box.

The Electronic Signature dialog box appears each time an operator performs an action that requires an electronic signature. If the tag associated with the action requires Perform Only signature, the Electronic Signature dialog box displays the Performed By section only, as shown in the following figure.

*Electronic Signature Dialog Box - Performed By*

If the tag associated with the action requires Perform and Verify signatures, the Electronic Signature dialog box displays the Performed By and Verified By sections, as shown in the following figure. When a user signs in the Perform By section, the Verify By section is always dimmed; when a user signs in the Verify By section, the Perform By section is always dimmed. Entering comments in the Verify Comment section is optional.



*Electronic Signature Dialog Box - Performed By and Verified By*

The following information explains each section of the Electronic Signature dialog box:

**Description Area** - describes the action to be signed for. This area is located at the top of the dialog box. If the action is a data entry, the description area includes:

- Fully-qualified name of the tag being changed.
- Tag's descriptor.
- Action performed; for data entry the description is "Change Value."
- Previous value.
- New value.

When a digital value changes, such as the current value of a digital tag or the auto/manual status of a tag, the appropriate labels, such as OPEN or CLOSE, display for the previous and new values.

If the action is a single alarm acknowledgement, the description area includes the node and name of the tag being acknowledged, and the tag's descriptor.

If the action is multiple alarm acknowledgement, the description area includes the nodes and names of all the tags being acknowledged, and each tag's descriptor.

**Performed By** - section of the Electronic Signature dialog box that displays when the tag is configured for electronic signature. If the tag is configured for Perform By signature, only this section of the Electronic Signature dialog box displays. If the tag is configured for Perform By and Verify By, both the Performed By and the Verified By sections of the Electronic Signature dialog box display.

If the tag is configured for Perform By signature only, once the operator enters his user name and password, and then clicks OK, the signature is validated, the value is written to the database, and the Electronic Signature dialog box closes.

If the tag is configured for Perform and Verify signature, once the operator enters his user name and password, and then clicks OK, the Performed By section of the Electronic Signature dialog box dims and the Verified By section activates.

**User Name** - name of the user performing the action or verifying the action. The name you supply here is your iFIX user name.

If the tag associated with the action allows continuous use, the user name of the continuous user automatically displays in this field in the Performed By section. The name of the continuous user is recorded from the last valid signature. You can enter a different name in this field. Refer to the [Allow Continuous Use](#) section for more information.

**Password** - password for the user performing the action or verifying the action. The password you supply here is the password assigned to your Windows or iFIX user account.

**Verified By** - optional section of the Electronic Signature dialog box that displays when the tag is configured with a signature type of Perform and Verify. After the Performed By signature is validated, the Verified By section activates and the Performed By section dims. Once the person who verifies the signature enters his user name and password in the Verified By section and then clicks OK, the signature is validated, the value is written to the database, and the Electronic Signature dialog box closes.

**Comment** - field available in both the Performed By and Verified By sections in which the operator or the person verifying the action enters comments. When the Perform By Comment Required option is enabled, text must be entered in the Comments field.

You can select or change a pre-defined comment from the drop-down list, or enter an original one in the text box. When the operator selects a pre-defined comment, it displays in the Comment field.

When entering text in the Comment field, you must enter less than 168 characters. However, when Perform By Comment Required is enabled in that block, you must enter more than 1 character but less than 168 characters.

In addition to entering text in the Comment field, you can also change the text of the predefined comment as it displays in the Comment field. Your changes do not alter the text of the predefined comment stored in the comment table.

If a comment table cannot be read for any reason, or if the application developer did not configure a comment table, the Predefined Comments drop-down list is dimmed.

For more information, refer to [Using Comment Tables](#).

## How the Electronic Signature Dialog Box Works

When the Electronic Signature dialog box appears:

1. The operator enters his:
  - User name
  - Password
  - Comment, optionally
2. The operator clicks OK.
3. If necessary, the person verifying the action performs steps 1 and 2.
4. The signature(s) are then validated and accepted or rejected.
5. If the signature(s) are accepted, the new value is written to the tag.

## How Do Data Links and Text Objects Behave at Run Time?

The behavior of data links and text objects configured for in-place data entry changes in run-time mode when the associated tag requires electronic signature. When the operator enters a value and presses Enter, the Electronic Signature dialog box appears.

## How Do Data Entry Objects Behave at Run Time?

The following list shows the data entry methods you can choose from the Data Entry Expert and how each method behaves when the associated tag requires electronic signature:

**Numeric/Alphabetic Entry** - When the operator enters a new value and clicks OK, the Electronic Signature dialog box appears.

**Slider Entry** - When the operator moves the slider bar and clicks OK or enters a value in the edit box and clicks OK, the Electronic Signature dialog box appears.

**NOTE:** When using the slider bar with electronic signature, you cannot also use the Write Continuously option. When the application developer chooses the Slider Entry option, he should clear the Write Continuously check box in the Data Entry Expert dialog box. If the Write Continuously option is left enabled, electronic signature is ignored at run time.

**PushButton Entry** - When the operator clicks a toggle button in the PushButton Entry dialog box, the Electronic Signature dialog box appears.

**Ramp Entry** - Each time the operator clicks one of the four ramp buttons, the Electronic Signature dialog box appears.

## Electronic Signature Examples

The following examples illustrate how the Electronic Signature feature works at run time. Each example assumes that George Clark is an operator with Perform By privileges and Thomas White is a supervisor with Verify By privileges. The examples used in this section examine these scenarios:

- Operator George Clark performs an action that he must sign for, but the action does not require verification. Refer to [Example 1: Perform Only Signature](#).
- Operator George Clark performs an action that he must sign for, and his supervisor, Thomas White, verifies that action. Refer to [Example 2: Perform and Verify Signature](#).
- Operator George Clark performs an action that requires a signature, but the value changes before he completes the signature. Refer to [Example 3: Value Changes During Signing](#).
- Operator George Clark performs an action and his supervisor, Thomas White, verifies that action. Each signer selects a predefined comment and enters an additional comment. Refer to [Example 4: Selecting and Entering Comments When Signing](#).
- Operator George Clark performs an action that requires a signature, but when he tries to sign for his action, he enters his password incorrectly too many times, and his account becomes disabled. Refer to [Example 5: Account is Disabled](#).

#### **Example 1: Perform Only Signature**

George Clark changes the value of a data link that uses the IFIX1\_PHARM\_HSM1\_TEMP\_SP tag as the data source. He changes the value from 10 to 20. This tag has been configured to require the signature of the performer only. When George changes the value and then presses Enter, the Electronic Signature dialog box appears, with the Performed By section displayed, as shown in the following figure.

*Perform Only Signature*

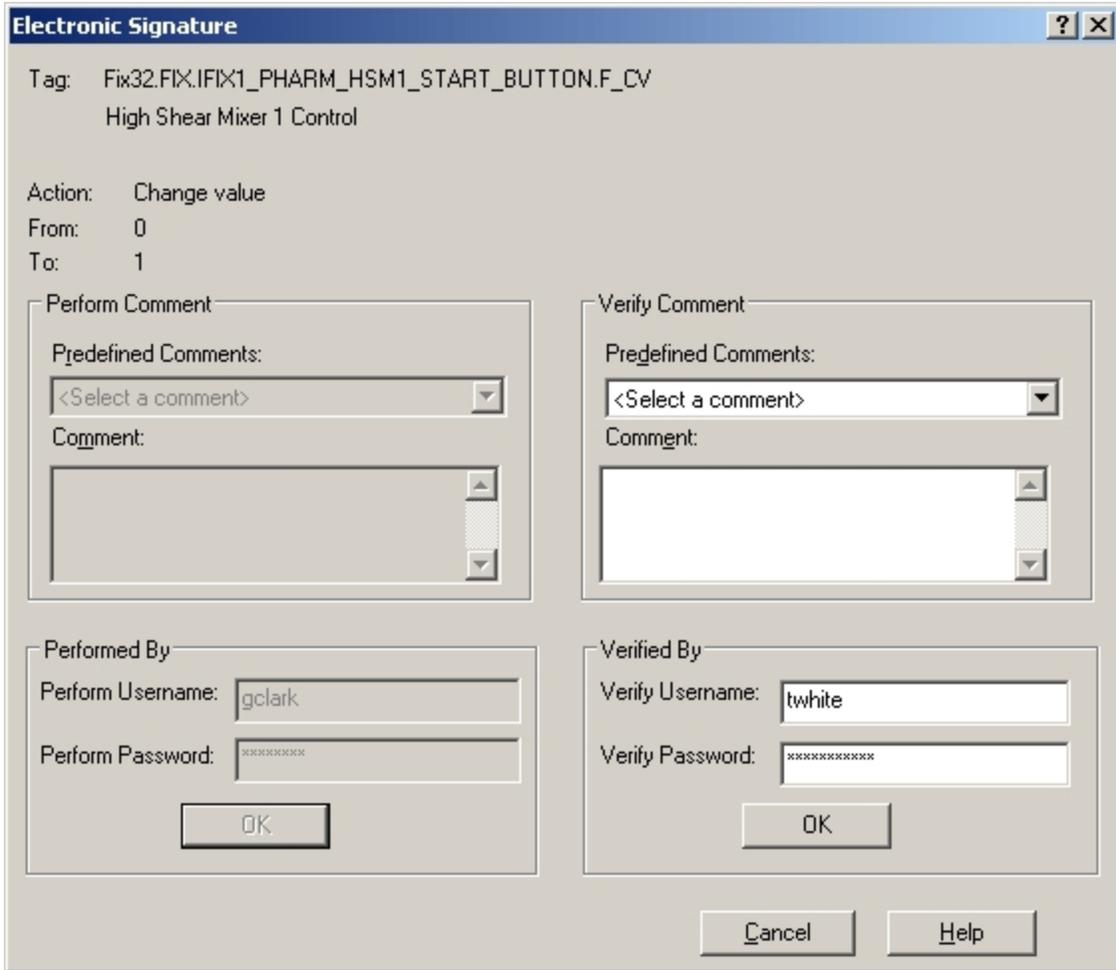
George signs for this action by entering his user name and password, and then clicks OK. Because the tag George signed for does not require a user to verify the action, his signature is validated, the value is written to the tag, the Electronic Signature dialog box closes, and the updated value displays in the data link. A message is written to the audit trail that details George's action.

### **Example 2: Perform and Verify Signature**

George Clark changes the value of a data link that uses the IFIX1\_PHARM\_HSM1\_START\_BUTTON tag as the data source. He changes the value from 0 to 1. This tag has been configured to require Perform and Verify signatures. When George changes the value and then presses the Enter key, the Electronic Signature dialog box appears, with the Performed By and Verified By sections displayed.

The Verified By section of the dialog box remains dimmed until George successfully enters his user name and password. When George clicks OK, his signature is validated, the Performed By section dims, and the Verified By section activates.

George's supervisor, Thomas White, enters his user name and password, as shown in the following figure. When Thomas clicks OK, his signature is validated, the value is written to the tag, the Electronic Signature dialog box closes, and the updated value displays in the data link. A message is written to the audit trail that details this action, including both George's and Thomas' signatures.

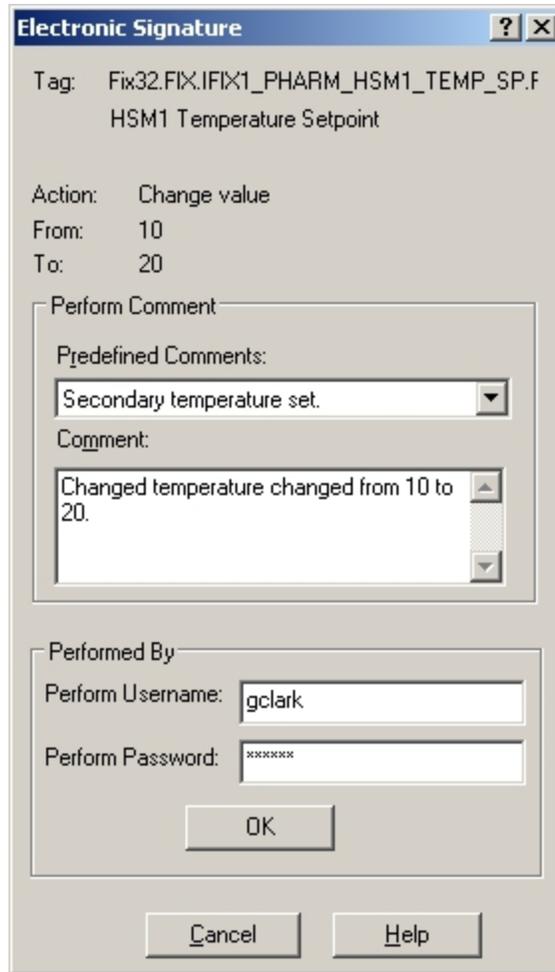


*Perform and Verify Signature*

### Example 3: Value Changes During Signing

George Clark changes the value of a data link that uses the IFIX1\_PHARM\_HSM1\_TEMP\_SP tag as the data source. He changes the value from 10 to 20. This tag has been configured to require Performed Only signature. When George changes the value and then presses the Enter key, the Electronic Signature dialog box appears.

When the Electronic Signature dialog box initially appears, the current value of the tag and the value George entered display. In this example, the current value of 10 is being changed to 20, as shown in the following figure.



When George signs for the action by entering his user name and password, and then clicks OK, the current value of the tag is double-checked. If the original value (10) differs from the real-time value of that tag, George's value is not written, and an informational message displays. This message indicates that the original value of the tag changed while George was signing for the action:

```
Error Number: -2147210972 (80042924)
The current value has changed. Unable to perform this signed write.
```

#### Example 4: Selecting and Entering Comments When Signing

George Clark changes the value of a data link that uses a tag that requires Perform and Verify signatures. Predefined comments are available to both signers, as indicated by the Predefined Comments fields. George selects a predefined comment and enters additional text in the Comment field. George signs and clicks OK.

When Thomas White verifies this action, he also selects a predefined comment and adds more text in the Comment text field, as illustrated in the following figure.

*Entering a Comment When Signing*

### Example 5: Account is Disabled

George Clark changes the value of a data link that uses the IFIX1\_PHARM\_HSM1\_TEMP\_SP2 tag as the data source. He changes the value from 10 to 35. This tag has been configured to require the signature of the performer only. When George changes the value and then presses the Enter key, the Electronic Signature dialog box appears.

George forgets his password and makes an incorrect guess. The following message displays:

```
Error Number: -2147210963 (8004292d)
Unknown user name or bad password.
```

George clicks the OK button and tries again. The bad password message displays for each subsequent incorrect attempt until he reaches the account lockout threshold, which, in this example, is set at three. On George's fourth incorrect attempt, the following message displays:

```
Error Number: -2147210967 (80042929)
Account currently disabled.
```

The application developer can change the text of the account disabled message. Refer to the [Account Lockout](#) section of the Getting Started chapter for more information.

## Using the Alarm Summary Object at Run Time

The Alarm Summary object supports the Electronic Signature option. Both single and multiple alarm acknowledgement is supported. Manual alarm deletion is also supported.

The operator can acknowledge one or more alarms using the Alarm Summary object with any of these conventional methods:

- Double-click a row.
- Select one or more rows and then press Enter.
- Use the Acknowledge Page command from the right mouse menu.
- Use the Acknowledge an Alarm command from the right mouse menu.
- Select the Alarm Summary object and press K on the keyboard.

Whenever the operator acknowledges an alarm for a tag configured to require electronic signature using one of these methods, the Electronic Signature dialog box appears.

The Electronic Signatures dialog box appears when the operator acknowledges multiple alarms. The dialog box displays the names and descriptions of all the tags being acknowledged by the operator. The alarms the operator acknowledges may have mixed signing requirements; some may require no signature, some may require Perform Only signature, and others may require Perform and Verify signatures.

When the operator acknowledges two or more alarms, the most restrictive signing requirements are enforced. The following table shows the signing requirements for the indicated conditions:

<b>Electronic Signature Signing Requirements</b>	
<b>How are tags configured?</b>	<b>Signing requirement...</b>
No tags require signature.	The Electronic Signature dialog box does not appear; no signature is required.
Some tags require no signature; some require Perform Only signature.	The Performed By section of the Electronic Signature dialog box appears.
Some tags require no signature; some tags require Perform Only signature; some tags require Perform and Verify signature.	The Performed By and Verified By sections of the Electronic Signature dialog box appears.

## Acknowledging a Single Alarm

When the operator acknowledges a single alarm for a tag that requires a Perform Only signature, the Performed By section of the Electronic Signature dialog box appears. When the operator acknowledges a single alarm for a tag that requires Perform and Verify signature, the Performed By and Verified By sections of the Electronic Signature dialog box appear.

The Electronic Signature dialog box for alarm acknowledgement is identical in layout and function to the dialog box for data entry. However, the description of the action displays as Acknowledge ALARM, as indicated in the following figure. The information displayed here includes the node name, tag name, and description of the tag whose alarm is being acknowledged.

The screenshot shows a dialog box titled "Electronic Signature" with a standard Windows-style title bar (minimize, maximize, close buttons). The dialog is divided into several sections:

- Tag Information:** "Tag: FIX.IFIX1\_PHARM\_BULKLEVEL" and "Bulk Material Mass".
- Action:** "Acknowledge ALARM".
- Perform Comment Section:** Contains a "Predefined Comments" dropdown menu with "<Select a comment>" selected, and a "Comment:" text area with a scroll bar.
- Verify Comment Section:** Contains a "Predefined Comments" dropdown menu with "<Select a comment>" selected, and a "Comment:" text area with a scroll bar.
- Performed By Section:** Includes "Perform Username:" with the text "gclark" and "Perform Password:" with masked characters "XXXXXXXXXX".
- Verified By Section:** Includes "Verify Username:" with the text "twhite" and "Verify Password:" with masked characters "XXXXXXXXXX".

At the bottom of the dialog are three buttons: "OK" (under the Perform section), "OK" (under the Verify section), "Cancel", and "Help".

*Acknowledge A Single Alarm*

## Acknowledging a Page of Alarms

When the operator acknowledges a page of alarms whose tags require Perform Only signature, the Performed By section of the Electronic Signature dialog box appears. When the operator acknowledges a page of alarms whose tags require Perform and Verify signatures, the Performed By and Verified By sections of the Electronic Signature dialog box appear.

The Electronic Signature dialog box for alarm acknowledgement is identical in layout and function as the dialog box for data entry. However, it contains a list of all alarms to be acknowledged in a scrollable area, as indicated in the following figure. The information displayed here includes the node name, tag name, and description of each tag whose alarms are being acknowledged.

**Electronic Signature**

Acknowledge alarms:

- FIX.IFIX1\_PHARM\_CIPLEVEL (Clean In Place Tank Level)
- FIX.IFIX1\_PHARM\_RECLAIMLEVEL (Reclamation Tank Level)
- FIX.IFIX1\_PHARM\_BULKLEVEL (Bulk Material Mass)

Perform Comment

Predefined Comments: <Select a comment>

Comment:

Performed By

Perform Username: gclark

Perform Password: XXXXXXXXXX

OK

Verify Comment

Predefined Comments: <Select a comment>

Comment:

Verified By

Verify Username: twhite

Verify Password: XXXXXXXXXX

OK

Cancel Help

*Acknowledge a Page of Alarms*

When the operator acknowledges a page of alarms, a separate message is sent to the audit trail for each alarm acknowledged. Refer to the [Electronic Signature Signing Requirements](#) table for a list of electronic signature signing requirements.

## Creating an Electronic Signature Audit Trail

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 he entered the signature, and why he entered the signature.

iFIX stores the electronic signature audit trail in a relational database. A relational database provides you with an open, secure storage solution you can query using established methods to produce reports and perform analysis and review. The relational database must be ODBC-compliant, such as Microsoft's SQL Server or Oracle.

**NOTE:** Microsoft Access is not supported in the electronic signature environment because it is not secure enough to ensure tamper-resistance.

Each time an operator signs for an action, a message is sent to the relational database containing all the elements of the signature, including:

- User name and full name of the person that performed the action.
- User name and full name of the person that verified the action.
- Description of the action.
- Time the action occurred.
- Name of the iFIX node where the user signed.
- User name and full name of the person logged in to the iFIX security system when the user signed.
- Optional comments entered by the performer and verifier.

Additionally, fields such as the name of the iFIX tag and the name of the SCADA node are included. You can also configure up to four user-defined fields that can be read from the tag and incorporated into the message.

## Configuring the Alarm ODBC Service

The iFIX Alarm ODBC Service inserts alarms, messages, and the electronic signature audit trail into an ODBC-compliant relational database. The data is parsed into a set of columns you can query to produce reports and perform analysis. You can configure the columns you want to include in your relational database table using the Alarm ODBC Configurator.

You can configure the Connection Lost Tag in the Alarm ODBC Service Configuration dialog box, which allows you to specify a database tag that is used to indicate a broken connection with the relational database.

You can also configure a temporary file to store alarms when the Alarm ODBC Service cannot connect to the relational database. The temporary file is encrypted. This feature prevents the loss of any electronic signature messages while the relational database is down or offline. If you do not want a temporary file, leave the Lost Connection File field blank.

Other iFIX alarm destinations, such as the Alarm History and Alarm File, also receive electronic signature messages. Due to space constraints, these destinations do not display every part of the signature, but they do show who performed the action, who verified the action, a description of the action and the time the action took place. These destinations do not represent the electronic signature audit trail, but can be used to quickly check on recent actions.

**TIP:** For the most reliable performance, run the Alarm ODBC Service on the SCADA nodes in your application, and not on the iClients.

Refer to the [Implementing Alarms and Messages](#) manual for more information on the Alarm ODBC Service and other alarm issues.

## Defining the Relational Database Columns

Several relational database table columns are included in the Alarm ODBC service for signed operator actions. These columns allow you to perform detailed database queries. The columns for signed operator actions are:

**Tag Description** - tag's description field as entered in the process database.

**Operator Login User Name** - user name of the person currently logged in to iFIX.

**Operator Login Full Name** - full name of the person currently logged in to iFIX.

**Performed By User Name** - user name of the person performing the action.

**Performed By Full Name** - full name of the person performing the action.

**Verified By User Name** - user name of the person verifying the action.

**Verified By Full Name** - full name of the person verifying the action.

**Performed By Comment** - comment supplied by the operator performing the action.

**Verified By Comment** - comment supplied by the user verifying the action.

**Message ID** - Globally Unique Identifier (GUID) that uniquely identifies each message.

For information on the other columns in the Alarm ODBC service, refer to the [Implementing Alarms and Messages](#) manual.

For examples of signed operation actions, refer to [Electronic Signature Examples](#).

## Sending Signed Messages to a Relational Database

If you have successfully enabled the Alarm ODBC Service and have configured your relational database, then each time an operator successfully signs for an action, a message is sent to the relational database. The message contains the following:

- Timestamp for the action.
- Name of the SCADA node.
- Fully-qualified name of the data source being changed (for data entry actions only).

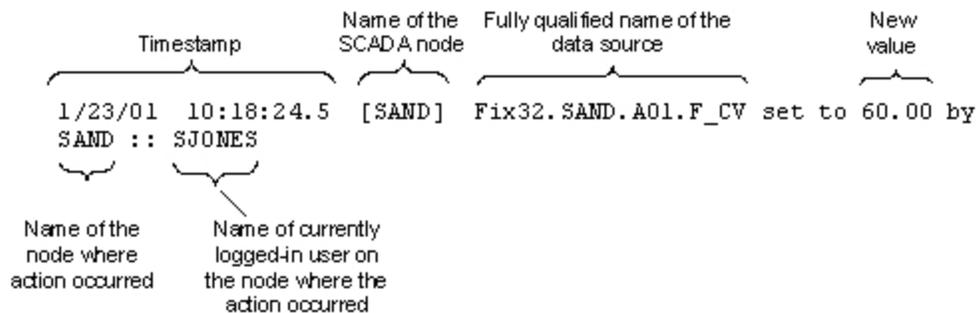
- Name of the tag whose alarm is acknowledged or deleted (for alarm acknowledgement and deletion actions only).
- Original value (for data entry actions only).
- New value (for data entry actions only).
- Operator's name.
- Optionally, the operator's comments.
- Name of the verifier and any related comments, if the tag requires verification.

When electronic signature is not required for the tag, the message sent to the relational database is the standard iFIX operator message containing the timestamp for the action, the user name of the logged-in user, and the new value.

Signed operator messages are sent to alarm areas configured for operator messages. Therefore, operator messages do not necessarily get sent to the same alarm area that the corresponding tag belongs to. Refer to the [Configuring Alarms](#) section of the Setting Up the Environment manual for more information.

### Example 1: Electronic Signature Not Required

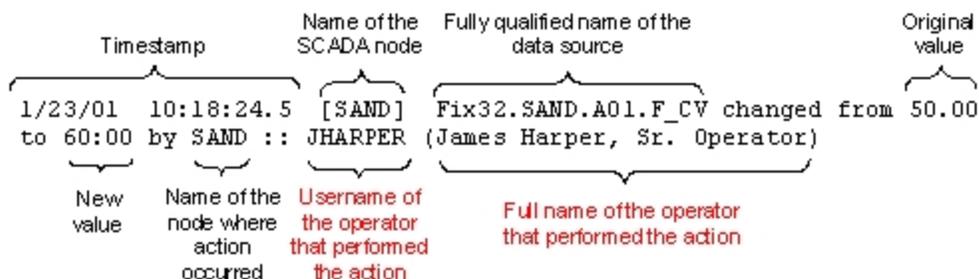
In this example, electronic signature is not required for the tag. This message is sent to the relational database when an operator changes the value of the tag:



In this message, SJONES represents the user name of the currently logged-in iFIX user.

### Example 2: Perform Only Signature Required

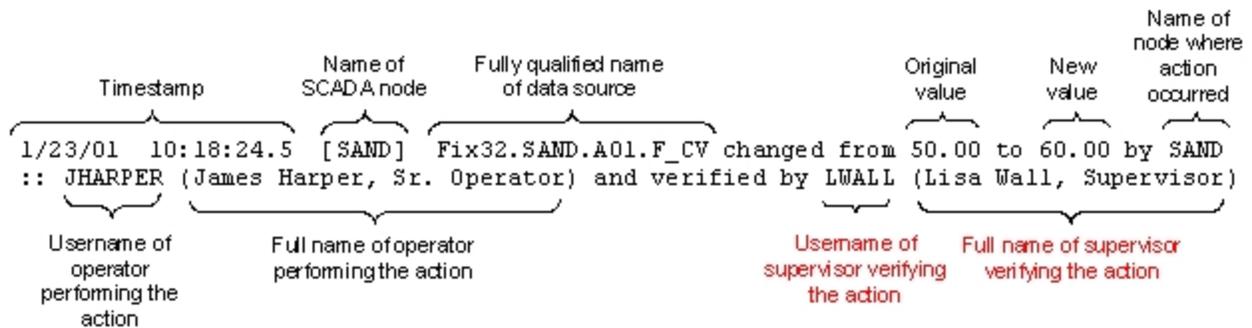
In this example, the tag requires a Perform Only signature. This message is sent to the relational database when an operator changes the value of the tag and signs for it:



In this message, JHARPER represents the user name, and James Harper, Sr. Operator is the full name of the operator that changed the analog setpoint value. Both the original value, 50.00, and the new value, 60.00, are clearly indicated.

### Example 3: Perform and Verify Signature Required

In this example, the tag requires both Perform and Verify signatures. This message is sent to the relational database when an operator changes the value of the tag, signs for it, and a supervisor verifies the change:



In this message, JHARPER is the user name, and James Harper, Sr. Operator is the full name of the operator that changed the analog setpoint value. LWALL is the user name, and Lisa Wall, Shift Supervisor, is the full name of the supervisor that verified this action. Both the original value, 50.00, and the new value, 60.00, are clearly indicated.

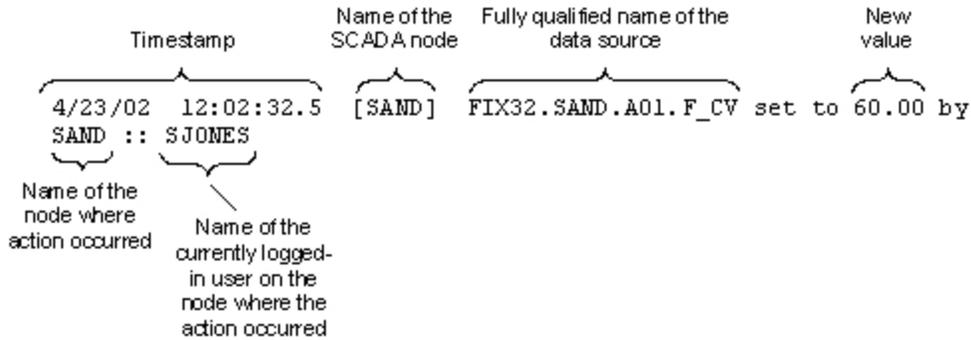
## Sending Signed Messages to Other Alarm Destinations

Each time an operator successfully signs for an action, a message is sent to the iFIX alarm system. You can view these messages in the Alarm History window, in the Alarm File, or on an alarm printer. The message contains the following:

- Timestamp for the action.
- Name of the SCADA node.
- Fully-qualified name of the data source being changed (for data entry actions only).
- New value (for data entry actions only).
- Operator's user name.
- Name of the tag whose alarm is acknowledged or deleted (for alarm acknowledgement and deletion actions only).
- User name of the verifier, if the tag requires verification.

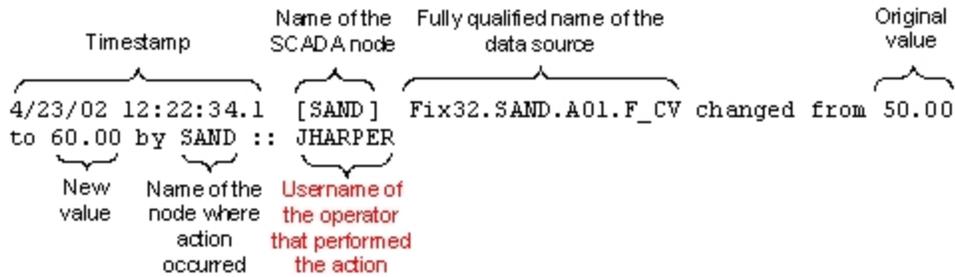
### Example 1: Electronic Signature Not Required

In this example, electronic signature is not required for the tag. This message is sent to the iFIX alarm system when an operator changes the value of the tag:



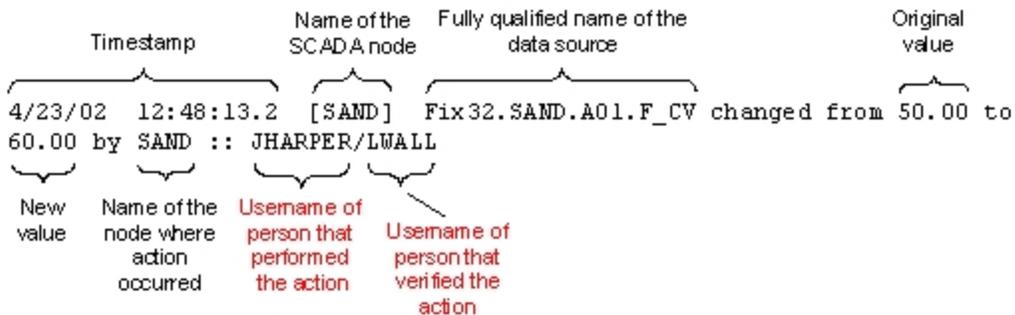
### Example 2: Perform Only Signature Required

In this example, the tag requires a Perform Only signature. This message is sent to the iFIX alarm system when an operator changes the value of the tag and signs for it:



### Example 3: Perform and Verify Signature Required

In this example, the tag requires both Perform and Verify signatures. This message is sent to the iFIX alarm system when an operator changes the value of the tag, signs for it, and a supervisor verifies the change:



## Using the Alarm Startup Queue Service

In the SCU, for the Alarm Startup Queue Service, do not clear the Summary alarms only check box in the Startup Queue Configuration dialog box. If you disable this option, duplicate messages may be sent to the audit trail. By default, the Summary alarms only check box is selected in the SCU.

## Using Electronic Signature in Scripts

This chapter provides examples of using electronic signature in scripts with the Electronic Signature object and with global subroutines and methods that support electronic signature. It includes the following sections:

- [Using the Electronic Signature Object](#)
- [Using Global Subroutines in Scripts for Electronic Signature](#)
- [Global Subroutines and Methods that Support Electronic Signature](#)
- [Examples: Using the Electronic Signature Object to Acknowledge Alarms](#)
- [Example: Using the Electronic Signature Object to Perform a Recipe Download](#)

## Using the Electronic Signature Object

You can create a script or another application that prompts the operator to enter an electronic signature using the iFIX Electronic Signature object. This allows you to:

- Integrate with badge readers and other signing mechanisms.
- Sign for writes to OPC data sources.
- Sign for other actions when writing to multiple data points, such as recipe download.

The Electronic Signature object is a COM object that implements the IESignature interface. The object can be instantiated by both VB/VBA and C/C++ code. You can call methods in the IESignature interface to:

- Determine if a tag requires a signature.
- Display the Electronic Signature dialog box.
- Validate a signature without displaying the Electronic Signature dialog box.
- Send a signed operator message to the audit trail.

**TIP:** Do not use scripts that use signing from the Scheduler. Signing will not work well from a background task. This is an important consideration when implementing scripts that call global subroutines, too.

For more information on using the Electronic Signature object, refer to examples provided later in this chapter and to the iFIX Automation Interfaces Help file.

## Using Global Subroutines in Scripts for Electronic Signature

When you use a global subroutine in a script that writes to the database, such as WriteValue, the Electronic Signature dialog box appears at run time if the tag is configured for signing. If one script uses two or more global subroutines that write to the database, the Electronic Signature dialog box appears for each tag that requires signature.

To prevent the Electronic Signature dialog box from appearing for each tag configured for signing, you can disable signing on one or more of the tags, or you can change the script.

**IMPORTANT:** If you do not use a global subroutine that supports electronic signature in your script, you must use the Electronic Signature object within the script to invoke the Electronic Signature dialog box.

## Global Subroutines and Methods that Support Electronic Signature

Several global subroutines and Alarm Summary object methods support electronic signature. If the data source that the global subroutine or method writes to requires electronic signature, the Electronic Signature dialog box displays. If the data source does not require signature, it is written directly.

Refer to the following sections for more details:

- [Global Subroutines and Methods that Support Electronic Signature](#)
- [Alarm Summary Object Methods That Support Electronic Signature](#)

## Global Subroutines Supported by Electronic Signature

The subroutines supported by electronic signature are:

- AcknowledgeAnAlarm
- EnableAlarm
- DisableAlarm
- OpenDigitalPoint
- CloseDigitalPoint
- ToggleDigitalPoint
- WriteValue
- RampValue
- OnScan
- OffScan
- ToggleScan
- ToggleManual
- SetAuto
- SetManual

**NOTE:** The AcknowledgeAllAlarms subroutine does not support electronic signature.

## Alarm Summary Object Methods That Support Electronic Signature

The following table indicates which Alarm Summary object methods support electronic signature.

<b>AlarmSummary Object Method</b>	<b>AlarmSummary Object Support Supports Electronic Signature?</b>
AckAlarmPageEx	Yes.
AckAlarm	No. Use the AcknowledgeAnAlarm global subroutine.
AckAlarmPage	No. Use the AckAlarmPageEx method.
AckAllAlarms	No.

## Examples: Using the Electronic Signature Object to Acknowledge Alarms

The following scripts provide examples of using the Electronic Signature object to acknowledge a list of alarms and a single alarm.

### Acknowledging a List of Alarms

The following sample script provides an example of using the Electronic Signature object, in this case to acknowledge a list of alarms:

```

Dim ESig As Object
Dim bNodeSignEnabled As Boolean
Dim bSigRequired As Boolean
Dim Value As Variant
Dim info As Integer
Value = 0
'Create the list of data sources
Dim DataSources As Variant
ReDim DataSources(2) As String
DataSources(0) = "Fix32.THISNODE.AI_NOSIG.F_CV"
DataSources(1) = "Fix32.THISNODE.AI_PERFORM.F_CV"
'Create the ESignature object
Set ESig = CreateObject("ElectronicSignature.ESignature")

'Check to see if node is enabled for electronic signature
ESig.IsNodeSignEnabled bNodeSignEnabled

If bNodeSignEnabled = True Then
    'Set data sources
    ESig.InitializeList DataSources

    'Is signature required for data sources
    '4 corresponds to ACK_OR_REMOVE_LIST
    ESig.IsSignatureRequiredForList 4, bSigRequired, info
    If bSigRequired Then
        ESig.GetSignatureAndWriteValue 4, Value
    Else
        'Acknowledge the alarms without signature
    End If
Else
    'Acknowledge the alarms without signature or warn node is not enabled for electronic signatures
End If

```

### Acknowledging a Single Alarm

The following sample script segment provides an example of using the Electronic Signature object, in this case to acknowledge a single alarm:

```

Dim ESig As Object

```

```

Dim bNodeSignEnabled As Boolean
Dim bSigRequired As Boolean
Dim Value As Variant
Dim info As Integer

Value = 0

'Create the ESignature object
Set ESig = CreateObject("ElectronicSignature.ESignature")

'Check to see if node has ESIG bit set
ESig.IsNodeSignEnabled bNodeSignEnabled

If bNodeSignEnabled = True Then
    'Initialize the object and set the data source
    ESig.Initialize ("Fix32.THISNODE.AI_PERFORM_VERIFY.A_CV")

    'Check if source requires signature
        '3 corresponds to ACK_OR_REMOVE
    ESig.IsSignatureRequired 3, bSigRequired, info

    If bSigRequired = True Then
        'Display the signature dialog
            '3 corresponds to ACK_OR_REMOVE
        ESig.GetSignatureAndWriteValue 3, Value
    Else
        'Acknowledge the alarm without signature
    End If

Else
    'Acknowledge the alarm without signature or warn node is not enabled for electronic signatures
    End If

```

## Example: Using the Electronic Signature Object to Perform a Recipe Download

The following sample script provides an example of using the Electronic Signature object, in this case to perform a recipe download:

```

Private Sub Rect1_Click()

Dim DownloadCommandLine As String
Dim return_value As Double
Dim CanDownload As Long
Dim RecipeCompStatusTag As Object
Dim ESignatureObject As Object
Dim bSignatureEnabled As Boolean
Dim bVerify As Boolean
Dim bAllowContinuousUse As Boolean
Dim bValidSig As Boolean
Dim PerformUser As String
Dim Performuserid As String
Dim PerformComment As String
Dim VerifyUser As String
Dim Verifyuserid As String
Dim VerifyComment As String
Dim bCheckStatus As Boolean

On Error GoTo RecipeDownloadError

' Check if the logged-in user has the privilege to download recipes
CanDownload = System.FixCheckApplicationAccess(52)

```

```

If CanDownload = 1 Then
    Set ESignatureObject = CreateObject("ElectronicSignature.ESignature")

    ' Check if the node supports electronic signature
    ESignatureObject.IsNodeSignEnabled bSignatureEnabled

    If bSignatureEnabled = True Then

        ' Get the signature
        bVerify = True
        bAllowContinuousUse = True
        ESignatureObject.GetSignature "Download Recipe testrcp", bVerify, bAllowContinuousUse, bValidSig, PerformUser,

    If bValidSig Then

        ' Download the recipe
        DownloadCommandLine = System.ProjectPath + "\RCPDOWN" + " /mtestrcp /iBATCH#1 /e"
        return_value = Shell(DownloadCommandLine, 0)

        ' Check the recipe completion status
        Set RecipeCompStatusTag = System.FindObject("fix32.thisnode.rcpcompstat.f_cv")
        bCheckStatus = True

        While bCheckStatus = True
            If RecipeCompStatusTag.Value <> 0 Then
                bCheckStatus = False
                If RecipeCompStatusTag.Value = 3 Then
                    ESignatureObject.SendSignedOperatorMessage "Downloaded Recipe testrcp", "", "", PerformUser, P
                    MsgBox "Recipe Downloaded Successfully!"
                Else
                    MsgBox "Recipe Download Failed"
                End If
            End If
        Wend
    Else
        MsgBox "No Signature, No Download"
    End If
Else
    MsgBox "Recipe download can only be done on a workstation that supports electronic signature"
End If
Else
    MsgBox "The logged-in user does not have sufficient security privilege to download recipes"
End If

Exit Sub

RecipeDownloadError:
    MsgBox (Err.Description)

End Sub

```

# Testing and Troubleshooting Electronic Signatures

This chapter contains information and suggestions about how to test and troubleshoot the Electronic Signature option. The following topics are described:

- [Disabling Signature for Testing Purposes](#)
- [Using a Local Computer with the Guest Account Enabled](#)
- [Changing Tag Values in the Database Manager Spreadsheet](#)
- [Determining if the Node is Enabled for Signing](#)
- [Determining if a Signature is Required](#)
- [Changing the Name of a Node](#)

## Disabling Signature for Testing Purposes

The application developer can disable the Electronic Signature option for testing purposes; this allows you to fully test an application without the need to repeatedly enter signatures.

To disable the electronic signature feature, create a user account that has the Electronic Signature - Bypass application feature assigned to it and then log in with that account. A corresponding user account with the Electronic Signature - Bypass application feature assigned to it must also exist on the SCADA node.

You can achieve the same effect by disabling security. However, this is not the recommended approach, since it leaves the system in a very vulnerable state.

**CAUTION:** Do not leave the Bypass account logged in, as it effectively disables the Electronic Signature option.

## Message Sent to Alarm System

When the Electronic Signature - Bypass application feature is enabled, a message is sent to the alarm system each time a tag configured for electronic signature is changed. If the Alarm ODBC Service is enabled, the message is also sent to the relational database; for example:

```
4/19/02 00:21.50.7[SAND] UNSIGNED WRITE ACCEPTED: FIX.AI2.F_CV - electronic signature bypassed by PSMITH.
```

The user name, in this example PSMITH, identifies the logged-in user on the node where the change was made.

When security is disabled, a similar message is sent each time a tag configured for electronic signature is changed; for example:

```
4/21/02 00:13.51.7[SAND] UNSIGNED WRITE ACCEPTED: FIX.AI2.F_CV - electronic signature bypassed - security disabled.
```

## Using a Local Computer with the Guest Account Enabled

If you use the Electronic Signature option on a local computer with the Guest account enabled, the security settings may not require password verification. In this situation, when you sign for an action, the system checks only for a valid user name. If you enter a valid user name, the signature is accepted even if you enter an incorrect password or leave the field blank.

To prevent this from happening, you can make the local computer a member of a domain. A domain is a group of computers that share a common directory on the same network. Your security accounts should be located on the domain controller, not on the local computer. You ensure the proper security and password settings when you use Windows security from the domain.

## Changing Tag Values in the Database Manager Spreadsheet

If a user changes a field of a tag that requires electronic signature directly in the Database Manager spreadsheet, that change is considered an unsigned write. Therefore, a user can change a field in a tag that requires electronic signature in the Database Manager spreadsheet only if one of these conditions is true:

- The tag is configured to accept unsigned writes.
- The currently logged-in user has Electronic Signature - Bypass application feature privilege assigned to his account. For more information on this application feature, see [Disabling Signature for Testing Purposes](#).

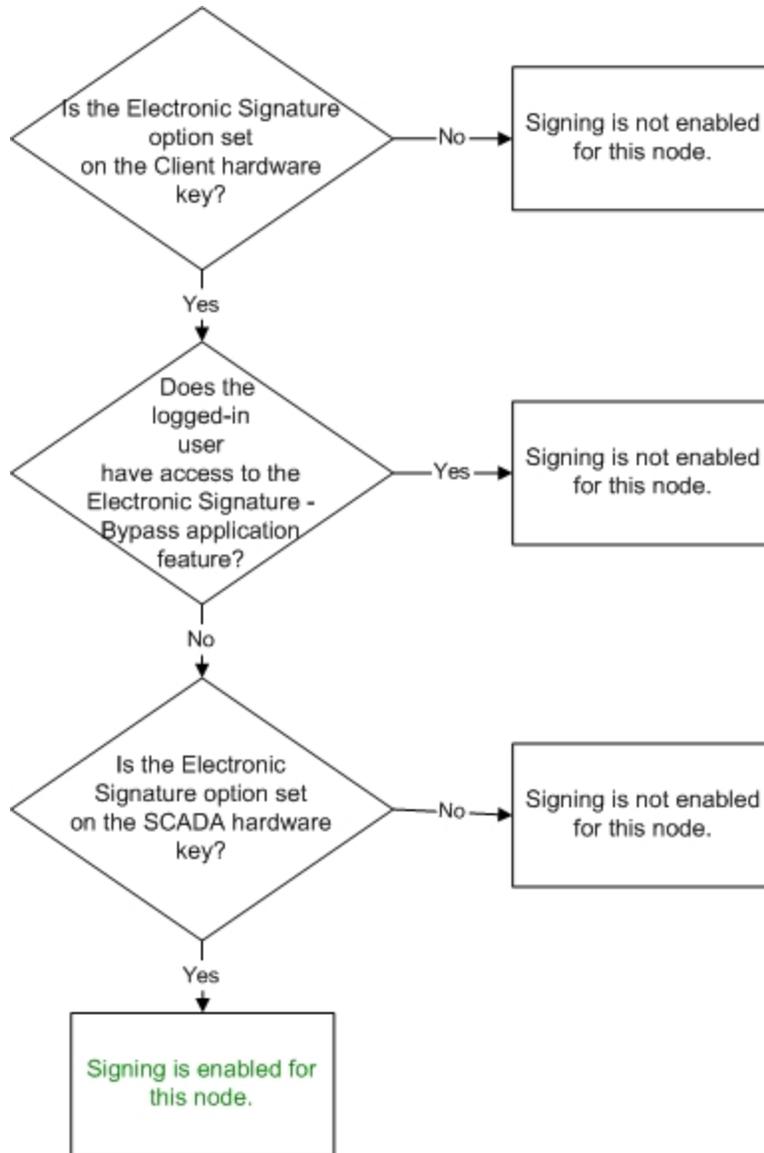
If neither condition is true, a message box appears when the user attempts to change a field of a tag that requires electronic signature. Here is an example of the text that appears in the message box:

```
[FIX:AI_1] Database block access requires electronic signature.
```

**NOTE:** Modifications to a tag made from the tag's configuration dialog box are always allowed.

## Determining if the Node is Enabled for Signing

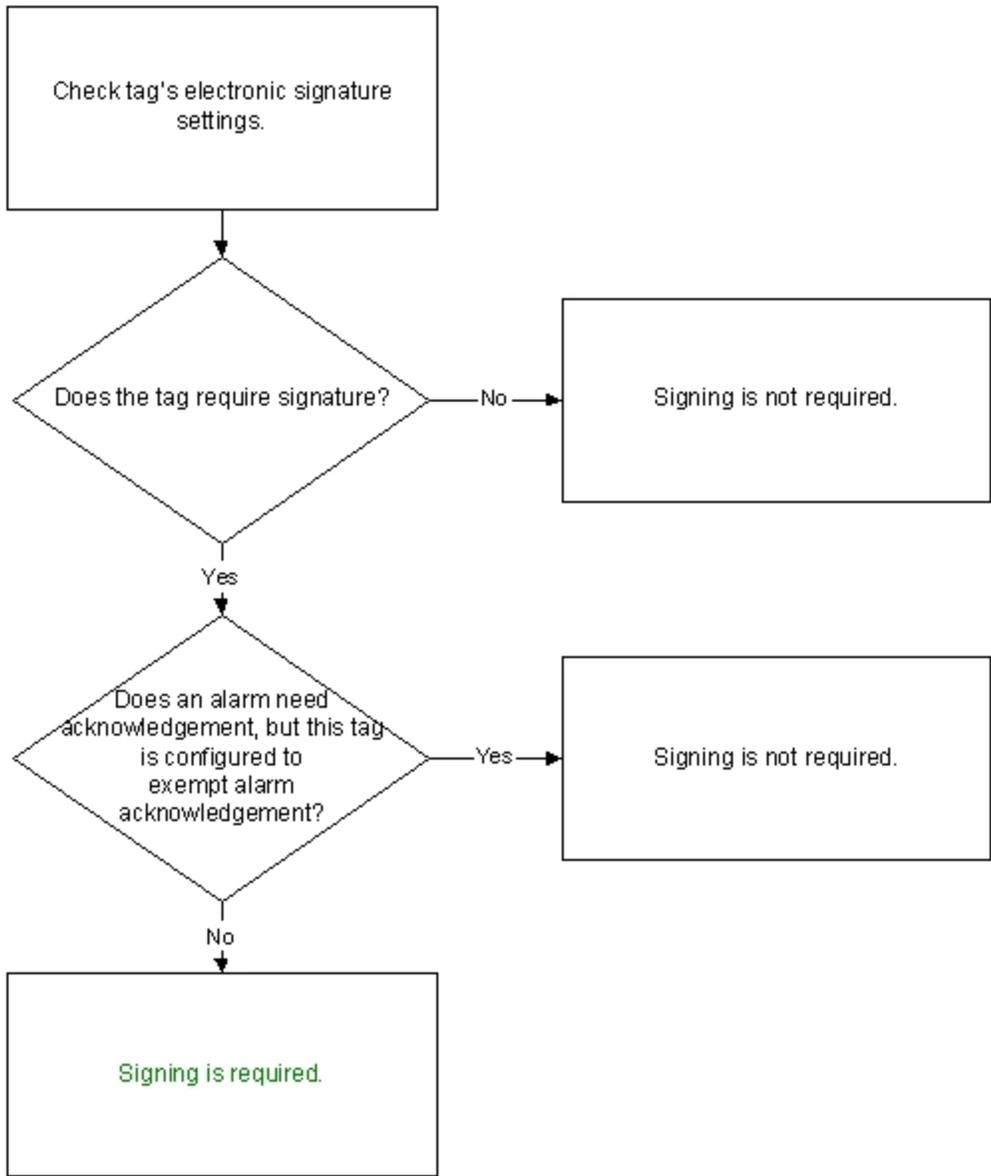
When an operator executes an action at run time using one of the signature-enabled user interfaces, such as the data link, these checks are performed to determine if the node is enabled for signing:



If the node is not enabled for signing, the action is executed without prompting for electronic signature.

## Determining if a Signature is Required

These checks are performed to determine if a signature is necessary for the data source if the node is enabled for signing:



## Changing the Name of a Node

If you change the name of a node where security is enabled, electronic signatures may not work as expected. This occurs because the system detects that security is disabled (when the security path is set to a folder other than the default, which is the C:\Program Files (x86)\GE\iFIX\Local folder). To resolve this issue, you must immediately disable and then re-enable security for the node and save the security configuration after you change the name of the node.

# Index

---

## 2

- 21 CFR Part 11 4
  - audit trail 33
  - understanding 4
- 21 CFR Part 11 Services 5

## A

- accepting and rejecting unsigned writes 15
- account lockout with electronic signature 7
- Acknowledge All alarms 17
- acknowledging alarms with electronic signature 17
  - all alarms 17
- adding comments to an electronic signature 2
- Alarm ODBC 33
  - changes for electronic signature 33
  - used with electronic signature 33
- Alarm Startup Queue Service 37
  - preventing duplicate messages in the audit trail 37
- Alarm Summary object 30
  - acknowledging alarms with electronic signature 30
- audit trail
  - configuring for electronic signature 32
  - preventing duplicate messages 37

## B

- behavior of unsigned writes 16

---

## C

- changing fields in database manager with electronic signature 44
- comment table 18
  - creating 18
  - deleting 18
  - renaming 18
- comment tables 18
  - creating for electronic signature 18
- configuring database columns for electronic signature 16
- configuring database tags for electronic signature 11
- configuring security for electronic signature 6
- continuous use 13
  - definition 13
- continuous use with electronic signature 13
- creating an audit trail for electronic signature 32
- creating comment tables for electronic signature 18

## D

- Data Entry Expert 17
  - defining operator limits 17
- database tags 11
  - configuring for electronic signature 11
- defining relational database columns for electronic signature 34
- disabled accounts with electronic signature 7

## E

- electronic signature
  - 21 CFR Part 11 Services 5

---

account lockout 7

acknowledging a page of alarms 31

acknowledging a single alarm 30

acknowledging multiple alarms 30

adding comments 2

audit trail 2

audit trail records 3

behavior of data entry objects at run time 24

behavior of data links and text objects at run time 24

calling methods in the IESignature interface 38

changes to the Alarm ODBC Service 33

changing fields in database manager 44

changing name of the iClient node 46

comment tables 18

configuring a comment table 18

configuring Alarm ODBC 33

configuring database tags for 11

configuring security 6

configuring spreadsheet columns for the database 16

conforming to 21 CFR Part 11 1

continuous use 13

creating an audit trail 32

creating comment tables 18

defining relational database columns 34

definition 2

deleting comment tables 18

determining a signed action 3

determining if a signature is required for the node 45

determining if node is enabled for signing 44

---

disabled accounts 7

disabling signature for testing 43

Electronic Signature dialog box 20

ensuring password verification 43

establishing security 5

examples of messages sent to relational database 34

exempt alarm acknowledgement 14

features and benefits 4

global subroutines supported 39

iFIX Screen Saver 10

implementing 5

implications of database writes 16

license and key checking 6

messages sent to relational database 32

methods supported 39

password 3

password expirations 7

Perform and Verify option 11

Perform Only option 11

performed by user 2

performing and verifying actions 2

recipe download sample script 41

renaming comment tables 18

restricting access from remote nodes 8

restrictive signing requirements 30

sending signed messages to a relational database 34

sending signed messages to the history file 36

signing examples 24

signing for your actions 24

---

- testing and troubleshooting 43
- tracking signatures 3
- tracking unsuccessful attempts to access iFIX 8
- understanding how security affects signing 8
- unsigned writes 15
- user name 3
- using at run time 19
- using multiple versions of iFIX in a network 10
- using operator limits with AO tags 17
- using other applications for signing 38
- using the Alarm Summary object 30
- using Windows locally 43
- using Windows user accounts 7
- verified by user 2
- writing scripts for signing 38
- electronic signature comments 18
  - free-form 18
  - predefined 18
- Electronic Signature dialog box 20
  - fields 20
  - how it works 23
  - Performed By and Verified By sections 23
  - Performed By section 20
- electronic signature examples 24
  - account is disabled 25
  - Performed By and Verified By 26
  - Performed By signature only 24
  - selecting and entering comments 25
  - values change during signing 24

---

- Electronic Signature object 38
  - recipe download sample script 41
  - using to prompt for signature 38
- Electronic Signature option 19
  - overview 1
  - using at run time 19
- establishing security for electronic signature 5
- exempt alarm acknowledgement with electronic signature 14

## F

- features and benefits of electronic signature 4

## G

- global subroutine 38
- global subroutines supported by electronic signature 39

## H

- history file 36
  - sending signed messages to 36

## I

- iClient node 46
  - changing name while using electronic signature 46
- IESignature interface 38
- iFIX Screen Saver 10
- implementing electronic signature 5

## M

- methods supported by electronic signature 39

---

## P

- passwords 2
  - expirations with electronic signature 7
  - used in electronic signature 2
- perform and verify signature 2
- perform only signature 2
- performing and verifying actions 2

## R

- relational database 34
  - columns for electronic signature 34
  - sending signed messages to 34
- restricting access from remote nodes 8

## S

- sample script 41
  - using Electronic Signature object to download a recipe 41
- security 6
  - configuring for electronic signature 6
- Security Synchronizer 8
- signing for your actions 24

## T

- testing electronic signature 42
- tracking electronic signatures 3
- troubleshooting electronic signature 43

## U

- understanding 21 CFR Part 11 regulation 4
- understanding how security affects signing 8

---

## unsigned writes

- behavior 16
- unsigned writes with electronic signature 15
  - accepting and rejecting 15
  - understanding 15
- user names used in electronic signature 2
- using comment tables 18
- using operator limits with AO tags 17
- using Security Synchronizer 8

## W

- Windows Security
  - disabling ability to change system time 7
- Windows user accounts 7
  - with electronic signature 7
- Windows XP 43
  - using locally 43
- WriteValue 38