Meridium APM Foundation

3.6.1.3.0

About This Document

This file is provided so that you can easily print this section of the Meridium APM Help system.

You should, however, use the Help system instead of a printed document. This is because the Help system provides hyperlinks that will assist you in easily locating the related instructions that you need. Such links are not available in a print document format.

The Meridium APM Help system can be accessed within Meridium APM itself or via the Meridium APM Documentation Website ([https://www.me-ridium.com/documentation/WebHelp/WebHelpMaster.htm](https://www.meridium.com/documentation/WebHelp/WebHelpMaster.htm)).

> **Note**: If you do not have access to the Meridium APM Documentation Website, contact [Meridium Global Support Services](#).

Table of Contents

About Core Analysis

The Meridium APM *Core Analysis* consist of a set of basic tools that allow you to view and manage the data in your system. These tools serve as the backbone on which the Meridium APM Framework application is built and represent the basic skill set with which you must be familiar before you begin using the other Meridium APM modules.

The Meridium APM Configuration Tools consist of the following components:

- [Queries](#)
- [Datasets](#)
- [Dashboards](#)
- [Graphs](#)
- [Reports](#)
- [R Scripts](#)
- [Strategy Rules](#)
- [Task Management](#)

Core Analysis System Requirements

The *Meridium APM Framework Tools* license is required to take advantage of Core Analysis functionality. In addition, your system must contain the basic architecture.

After you have installed and configured the basic system architecture, you will need to perform some configuration tasks specifically for the Core Analysis tools.

The Meridium APM Framework contains a feature that allows you to share dashboards via email. This feature is supported only on specific browsers and mobile operating systems.

Overview of Dashboards

A *dashboard* is a Meridium APM Catalog item that can be used to display summary information about the data in your Meridium APM system and provide a way to navigate to other areas in the Meridium APM Framework. Dashboards are stored in the Meridium APM Catalog, and they are displayed on the **Dashboard** page. The content of each dashboard is driven by queries that define what should be displayed on the dashboard. Throughout this documentation, we use the term *dashboard* to refer the dashboard Catalog item and its underlying queries that are displayed on the **Dashboard page**.

Dashboard Catalog items store *widgets* that are used to configure dashboard sections, their configuration options, and default layouts. The term *widget* is used in the Meridium APM documentation to refer to the code that determines the content that is displayed in each section on a dashboard.

Depending upon your security permissions to the Catalog folders in which dashboards and their underlying queries are stored, you can create new dashboards, modify existing dashboards, or perform various operations on existing dashboards. The Meridium APM product includes a set of baseline dashboards that are specific to certain modules and work processes.

Supported Environments for Viewing Dashboards Online

Dashboards can be [shared with others via a link in an email message](). Clicking the link opens the dashboard in a web browser.

> **Note:** The PC or mobile device you use to view the dashboard link must be connected to a network that can access the Meridium APM Application Server.

For PCs, the following web browsers are supported:

- Microsoft Internet Explorer 10 (and above)
- Google Chrome (which updates itself automatically)
- Apple Safari 6 (and above)

For mobile devices, the following operating systems are supported. Note that certain dashboard features, such as the button and the button, are not available when viewing shared dashboards on a mobile device.

- Google Android 4.0 (and above)
- Apple iOS 6.1 (and above)
- Microsoft Windows 8 (and above)
- Microsoft Windows Phone 8 (and above)

For mobile devices, the default browser depends on the device you are using. In addition, on many devices, you can install different web browsers. For this reason, support is listed by operating system, rather than by specific web browsers.

Overview of Datasets

A *dataset* is a fixed set of information that is stored as a Catalog item. Datasets contain information that can be viewed and modified in a grid, where columns and rows form intersecting cells that store data. Datasets can be used for storing static data and will often represent data not stored in records that exist in the Meridium database. While queries return the most current information from the database, a dataset displays the same information every time you view the dataset. For example, you might use a dataset to store information from an external system so that the data can be used to create reports and graphs in the Meridium APM Framework application.

Datasets can store certain types of data. When a dataset is created, a data type is defined for each column to indicate the type of data that can be stored in cells in that column. When you create or modify a dataset, if you attempt to enter data in a cell that is inappropriate for the data type defined for that column, an error message will be displayed, and the data will not be retained. The following types of data can be stored in a dataset:

- **Character:** Stores alphanumeric data. Cells in columns of this data type will appear as text boxes, where you can type letters, numbers, and symbols.
- **Numeric:** Stores numeric data. Cells in columns of this data type will appear as text boxes, where you can type numbers.
- **Date:** Stores date and time data in the format MM/DD/YYYY HH:MM <A.M. or P.M.>. Cells in columns of this data type will display an arrow that provides access to the Calendar feature, but you can also type dates in date cells.
- **Logical:** Stores the value *True* or *False*. Cells in columns of this data type will contain a check box that you can select to indicate *True* or clear to indicate *False*.

Using the **Dataset Builder**, you can create new datasets by importing a Microsoft Excel file (.XLS), a text file (.TXT or .CSV), or an XML file (.XML) or by creating a dataset and entering data directly into the **Dataset Builder**.

After a dataset exists, you can view and modify it on the **Dataset Editor** page, where you can also delete the dataset or use it to create graphs or reports. The **Dataset Editor** page also provides access to the **Dataset Export Builder**, which you can export a dataset to a Microsoft Excel, text, or XML file.

Example of a Dataset

To make this documentation easier to follow and to help explain how datasets can be imported and exported, in this topic we provide an example of a dataset. We will refer to this example throughout the dataset documentation as necessary.

In the following image, the **Dataset Editor** page displays the *Entry Log* dataset. This data-set stores information about entries into a building, including various personnel inform-ation and whether or not each entry was authorized.

For the purposes of this example, we will assume that this information is collected by a security system and stored in a database that is external to the Meridium APM database.

Note that the example contains the following columns of information:

- **ENTRY:** Uses the Numeric data type and stores a value representing the order in which the entries are logged. The example includes seven separate entries by four people, and every entry has a unique number.
- **ID:** Uses the Numeric data type and stores an identifying number for the person associated with each entry. These values can be repeated, since one person may enter the building multiple times.
- **NAME:** Uses the Character data type and stores the name of each person asso-ciated with each entry. These values can be repeated, since one person may enter the building multiple times.
- **DATE:** Uses the Date data type and stores the date and time at which each entry was made. This column will always have unique values, since only one person can enter the building at a time.
- **AUTHORIZED:** Uses the Logical data type and stores the value *True* if the entry was authorized or *False* if the entry was unauthorized. An entry could be unauthorized for a number of reasons, such as user errors in entering information. This column will have repeated values, since multiple authorized and unauthorized entries will exist.

- **notES:** Uses the Character data type and stores any notes about the entry. In this example, Entry 2 contains a note explaining that the entry was unauthorized due to an Unauthorized Entry Code.

Overview of Graphs

Graphs provide a visual representation of data. Graphs can be created and modified via the **Graph Viewer** page but are used through the Meridium APM Framework in both Core Analysis and in other modules. For example, the following image shows the *Analog Calibration Error Graph*, which is included in the Calibration Management module.
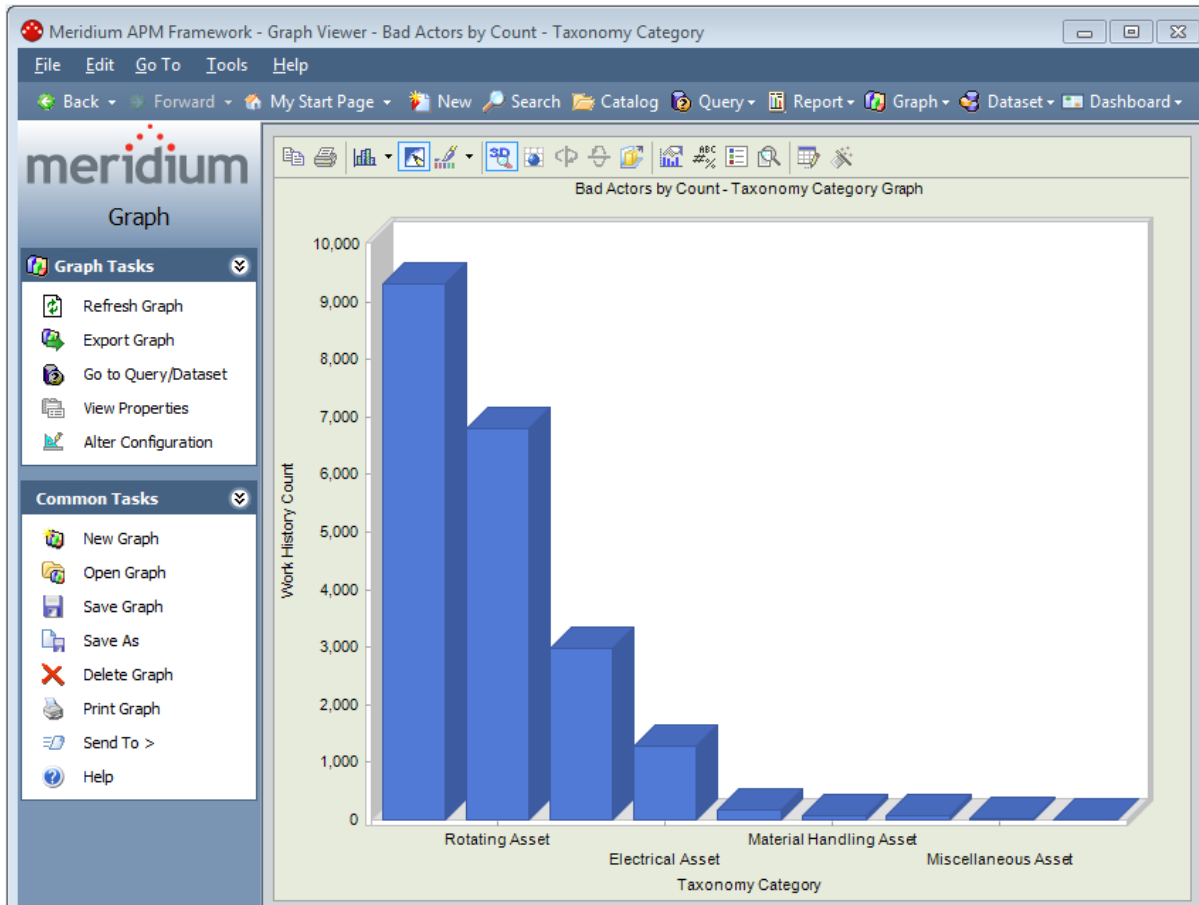
You can create graphs based on a query or dataset. When you create a graph, you will select the type of graph that you want to build and the information that you want to display. To create a graph, at least one field in the underlying query or one column in the underlying dataset must be *numeric*. For example, if a query returns the Equipment ID and number of failures associated with all pumps in production, you could display the Equipment IDs in a graph to provide a visual representation of the number of failures per pump.

Note that any values that you want to see in your graph must be returned by the query or included in the dataset on which that graph is based. Any changes that you make to the underlying data will be reflected in the graph. For example, if you build a graph based on a query that returns the repair costs for each pump in your system and then create a new pump record, the next time you view the graph, the data for the additional pump will appear.

Note: If you *remove* from the underlying query or dataset a field that is included in the graph (e.g., you remove the field on which the x-axis is based), the next time that you try to open the graph, an error message will appear, indicating that the information is not valid.

After a graph has been created, it is displayed on the **Graph Viewer** page (shown above), where you can view and modify areas on the graph.

Types of Graphs

When you create a graph, you will need to select the type of graph that you want to create and then specify the fields that you want to include in the graph. The following table lists each graph type and describes how it can be used.

Graph Type	Useful for...
Area	Identifying trends by tracking a series of values measured against a set scale. The information in an area graph is displayed in linear form.
Area Curve	The same purpose as an area graph except that the information in the graph is displayed in a curved form (non-linear).
Bar	Comparing a set or series of values to the same scale or unit of comparison or over time. The information on a bar graph is represented by vertical or horizontal bars.
Bubble	Comparing a set of two values, where one of the values specifies size or volume. The data that is displayed on a bubble graph is represented by bubble shapes.
Candlestick	Displaying statistics over a measured amount of time (e.g., stock prices). The incremental data is represented by a series of open or closed bars with a trailing line to indicate the lowest and highest values reached within the specified amount of time (e.g., a single day).
Cube	Comparing a set or series of values to the same scale or unit of comparison or over time. The data that is displayed on a cube graph is represented by cubes.
Curve	Identifying trends by measuring and comparing a series of values. The data that is displayed on a curve chart is represented by a line that curves to indicate changes in the data.
High-low-close	Displaying statistics over a measured amount of time (e.g., stock prices). The incremental data is represented by a series of vertical bars, whose lowest, highest, and final values (reached within the specified increment of time) are represented by a horizontal line. Note that to use this graph type, the query or dataset must contain at least one character and *three* numeric fields (i.e., highest, lowest, and closing values).
Lines	Identifying trends by comparing a series of values. The data that is displayed on a Line graph is represented by a line whose peaks and crevices indicate changes in the data.

Graph Type	Useful for...
⊪⊪ Open-high-low-close	The same as a high-low-close graph with one distinction: in addition to displaying the highest, lowest, and final values, an open-high-low-close graph also displays the *initial* values (reached within the specified increment of time). This is useful for determining trends within the data and over time. Note that to use this graph type, the query or dataset must contain at least one character and *four* numeric fields (i.e., the opening, highest, lowest, and closing values).
📊 Pareto	Combing different gallery styles to enhance the effectiveness of your chart. For example, you can use a pareto graph to compare the number of failures over time from 1970 to 2002. The y-axis would represent the number of failures, and the x-axis would represent the year.
⣿ Scatter	Comparing a series of X and Y coordinates and for identifying relationships between values in an X, Y coordinate. For example, you could use a scatter graph to view the cost of repair by failure date for a group of equipment.
⌐ Step	Identifying trends by measuring and comparing a series of values. The data that is displayed on a step graph is represented by a line, whose stair-like shapes indicate changes in the data.
◎ Dough-nut	Comparing portions of the graph to one another. For example, you could use a doughnut graph to determine which manufacturer's centrifugal pumps are most active. Note that you can also display an additional doughnut graph to compare to one another.
◕ Pie	Comparing a portion of the graph to the whole. For example, you could use the pie graph to determine the percentage of energy that is used by a particular piece of equipment or group of equipment.
▲ Pyramid	Comparing portions of the graph to one another and portions of the graph to the whole. For example, you could use a pyramid graph to compare the failure rates of multiple types of equipment.

Graph Type	Useful for...
🌟 Radar	Displaying multi-source values as they relate to one another. For example, you could use a radar graph to compare your allocated versus budgeted spending, where you can also view the level of spending per expense category (e.g., marketing and salaries). The data is a radar graph is displayed in a plot that resembles a spider-web.
📊 Gantt Chart	Comparing a set or series of values to the same scale or unit of comparison or over time. You can use this type of graph to display project plans and to show schedules for tasks or items in relation to one another. Note that this type of graph can have only one numeric column.
✅ Con-tour*	Representing a large number of values visually against each other on a set scale and for noting trends. For example, you could use a contour plot to view corrosion thickness readings, where the series on the graph would consist of different areas within a tank. The readings would be plotted on the graph, providing a visual representation of the corrosion rate.
📀 Sur-face*	The same as a contour plot except that the surface plot presents a two-dimensional view of how the surface of the item that is being displayed on the graph *looks*. For example, you could use a surface plot to view corrosion thickness readings.

* Because this type of graph relies on shading that represents values, you should have a minimum of 15 values in a series for the data to appear as it is intended. Note also that this type of graph requires at least two series and two numeric fields.

Overview of Queries

A *query* is a tool that you can use to retrieve information from the database. Using queries, you can construct a question or request of the database and then view the answer (i.e., the query results) on the screen. The records that are returned by the query can include fields from one query source, such as a family or another query, or several related query sources. After you construct a query, it can be saved for future use.

You can think of a query as a set of criteria that helps you define the specific information you want to retrieve from the database. For example, consider a query on the Pump family where Asset ID, Asset Manufacturer, and Asset Installation Date are the fields selected for inclusion in the output. Instead of using this query to return ALL the Pump records in the database, you might want to return only the Pump records where the Asset ID, Manufacturer, and Asset Installation Date match certain criteria. For example:

- You might apply criteria to the query to limit the output to pumps installed after a particular date.
- You could define criteria so that the results would show only pumps manufactured by a particular manufacturer.

So, using this example, your overall query criteria is that you want to see the pump ID, manufacturer, and installation date for all pumps that were installed after a particular date and manufactured by a particular manufacturer.

After you learn how to create basic queries, you can expand that knowledge to create more complex queries. The more you learn about creating queries to retrieve detailed information from the database, the better you can incorporate these queries into other areas of Meridium APM. For example, you can create queries on which to base graphs and you can configure queries to generate detailed reports.

You can construct queries using:

- The **Query Builder**, which takes you step-by-step through the process of adding query sources and fields to a new query.
- The query design, which is intended for intermediate users and gives you more flexibility in creating your query.
- The SQL code, which is intended for advanced users who understand how to construct SQL queries from scratch.

About Running Queries

Queries are available in many places throughout the Meridium APM Framework application. In some cases, you might create a query manually and save it for future use. In other cases, you might run a query that another user constructed or that is delivered with the baseline content.

When you run a query, you might see the results on the **Query Builder** page, which will make it obvious that you are looking at query results. For instance, if you open a query from the Catalog or click a Home Page hyperlink that references a stored query, the results will be displayed on the **Query Builder** page.

In other cases, however, you might run a query and see the results in a different format on a different screen. In these cases, you might not notice at first glance that you are looking at query results. Inspection Management, for example, provides a customized workflow that allows you to execute queries that return specific information, such as equipment that can have bundle inspections. In addition, searches provide a customized form where you can search the database for records in specific families that meet specific criteria. In other words, performing a search is simply another way of querying the database to return a subset of records.

When a query is created, it can be configured to return raw data stored in the database or reformatted data. This means that you might run different queries that return the same data in different formats. In addition, queries can be configured to prompt you for information before returning any results.

Queries are not stand-alone tools; they can be incorporated throughout your entire Meridium APM system.

About Query Privileges

Some query options will behave differently based upon the query privilege setting in your Security User record (i.e., the value in the Query Privilege field). The following sections describe how the queries will behave when you are logged in as a Security User with any of these query privilege settings. These sections describe only the options that behave differently. You can assume that any option that is *not* mentioned here will behave as described elsewhere in the documentation.

Behavior for unrestricted users

If a Security User is designated as *unrestricted* (i.e., in the Query Privilege field in the Security User record, the value is set to *Unrestriced*, or the field is empty), no query options will behave differently. Instead, queries will function exactly as described elsewhere in the documentation.

Behavior for users restricted by the time-out limit

If a Security User is designated as *restricted by the time-out limit* (i.e., in the Query Privilege field in the Security User record, the value is set to *Restricted By Timeout Limit*), the user will be able to create and modify existing queries. Before they can save the new or modified queries, however, they will first need to run the query. When the query runs, if a query timeout limit has been specified in the database, the Meridium APM system will allow the query to run until the specified query timeout limit has been met. After the timeout limit is reached:

- If the query has *not* returned results, a message will appear, indicating that the query cannot be saved, and the save options will remain disabled.
- If the query has returned results, the save options will be enabled so the Security User can save the query.

More specifically, while the results may not be displayed (depending upon the time-out setting), the following links will be disabled on the **Common Tasks** menu on the **Query Builder** page after you run the query:

- Save Query
- Save As

In addition, the following links will always be disabled on the **Common Tasks** menu on the **Query Builder** page when you are viewing either the query design or the SQL code for a query:

- Save Query
- Save As
- Delete Query

All the other Query Tool options will behave as described elsewhere in this documentation.

Behavior for users with only execute privileges

If a Security User is designated as *execute only* (i.e., in the Security User record, the Query Privilege field is set to *Execute Only*), the user will be able to view query results but not create or modify queries. This means that when the query results are displayed, these Security Users will not be able to access the query design or SQL code.

More specifically:

- On the **Query** submenu on the Meridium APM Framework toolbar, the **Open Existing** option will be the only available option. The **Build Query** option will be hidden.
- On the **Query Tasks** menu, the **Run Query** option will be the only available option. All other options will be hidden.
- On the **Common Tasks** menu, the following options will be available:
  - **Open Query**
  - **Print Query**
  - **Send To**
  - **Help**

All the other options on this menu will be hidden.

About Query Types

Two main categories of queries are available in Meridium APM: *data queries* and *action queries.* Each category can be further divided into specific types.

- **Data Query:** Retrieves data and displays the results on the screen. You can use two types of data queries:
    - [Select Query](): The most basic type of query, which returns rows of information in a columnar format.
    - [Crosstab Query](): A query that uses a grid in which one field is the row and another field is the column. The intersection of the row and column can be an aggregate of the two fields or an aggregate of a third, linking field.

    > **Note:** All users can create data queries using any family on which they have at least View privileges.

- **Action Query:** Modifies information in the database and does not return results to the screen. You can use three types of data queries:
    - [Update Query](): Modifies data in the database.
    - [Append Query](): Adds a group of records from one or more families to one or more other families.
    - [Delete Query](): Deletes rows of data.

    > **Note:** Only Super Users and members of the MI Power User Role Security Group can create action queries. All users can *run* action queries, provided that they have been granted permission to access the Catalog folder in which they are stored. A user's ability to perform the action (e.g., delete) associated with the query is controlled by family-level privileges.

When you create a query, the initial query type is set to *Select* by default. After you have created the basic query, however, you can modify the query type if needed using the **Options** link on the **Query Tasks** menu.

Select Queries

A *Select query* simply returns a list of records that belong to one or more specified families and match your query criteria. For example, you might want to return a list of:

- All the records in the Equipment family.
- All the records in the Functional Location family that contain a certain value in the Category field.
- All the Equipment records that are linked to a Functional Location records that contain a certain value in the Category field.

The results of a Select query can provide you with a comprehensive or custom view of the data that exists in the families in your database, allowing you to see only the records and fields that you need to see. Creating a Select query is also often the first step in a process that involves using the query results to perform a certain task. For example, you might create a Select query and then use those results to build a report or create a graph.

To create a Select query, you can:

- Use the **Query Builder** to build the query based on a family.
- Use the **Query Builder** to build the query based on an existing query.
- Design the query from scratch.

Crosstab Queries

A *Crosstab query* lets you group data into categories, where a category is determined by a value that exists in multiple fields across multiple families in the database. In the results of a Select query, each field appears in a column, providing a simple list of data. The results of a Crosstab query appear in a grid. In other words, a Crosstab query presents the same information as a Select query but in a different format. The format that you choose will depend on the type of information that is returned by the query and how you want to view it.

For example, suppose you have a family called Pumps, which stores data on Pump Location, Pump Manufacturer, and Pump Failures. If you queried the family and included in the output the location, manufacturer, and failures fields:

- A Select query would display the results like this:

Pump Location	Pump Manufacturer	Pump Failures
Zone 1	ACME	3
Zone 1	SUPER	5
Zone 2	ACME	4

...where each field appears as a separate column of information.

- A Crosstab query, on the other hand, might display the results like this:

	ACME	SUPER
Zone 1	3	5
Zone 2	4	NULL

...where locations are the rows and manufacturers are the columns.

In this example, you can see that ACME is the manufacturer of multiple pumps. The manufacturer, therefore, represents the category by which you want to display the remaining data (pump location). Each column in the results represents a separate value within the same category. So, in this example, ACME and SUPER are different types of manufacturers within the manufacturer category.

Delete Queries

A *Delete query* deletes from the database records that meet the criteria in the query. Delete queries delete *entire records*, not just individual fields in records. A Delete query is similar to a Select query in that it retrieves records from the database that match the criteria defined within the query. The difference is that instead of displaying the results, the results are deleted from the database.

If you need to delete many records at once, this type of query can save you time as it allows you to delete records in batch rather than individually. For example, suppose you created various records in Family A and later determined that those records were invalid. You could create a Delete query to delete all the records in the family, based upon the desired criteria, such as a value in the record, the creation date, or the user who created them.

Only Super Users and members of the MI Power User Role Security Group can create Delete queries. All users can *run* Delete queries, provided that they have been granted permission to access the Catalog folder in which they are stored. To delete the records returned by the query, a user must have Delete privileges on the family to which those records belong.

> **Note:** Deletions performed by a Delete query cannot be undone.

Update Queries

An *Update query* modifies the records that match the criteria that you have specified in the query. An Update query makes global changes to a group of records belonging to one or more families. An Update query is similar to a Select query in that it retrieves records from the database that match the criteria defined within the query. The difference is that instead of displaying the results, the results are modified according to the criteria defined in the query.

For example, if an equipment manufacturer name has changed, you would need to update all the existing records for equipment made by that manufacturer so that the value in the Manufacturer field is the new name of the manufacturer. Instead of manually modifying these values, you could create an Update query to modify all the records for this family.

Only Super Users and members of the MI Power User Role Security Group can create Update queries. All users can *run* Update queries, provided that they have been granted permission to access the Catalog folder in which they are stored. To update the records returned by the query, a user must have Update privileges on the family to which those records belong.

> **Note:** Modifications made by an Update query cannot be undone automatically. If you need to undo your changes, you will need to create another Update query or modify the records individually.

Append Queries

An *Append query* allows you to add a group of records from one or more families to another family. In other words, an Append query lets you:

- Use a predefined [Select query](#) to find records in one or more families that you want to add to another family.

- Take the records that are returned and *add* them to another families, thereby *creating* new records in those families.

For example, you might create a set of records in one family that are the same or similar to a set of records that you also need to create in another family. Or you might decide that you want to consolidate records and instead of storing records that currently exist in multiple families store them in a single family. In either of these cases, you could use an Append query to find the records that currently exist in one family and add them to another family. In cases where you are using an Append query to *move* records, you might also want to use a Delete query to delete the records from the original family after they have been moved.

In an Append query you will define field mappings to map fields in the source family to fields in the target family. Only fields that are included in the Select query can be mapped to the target family, so you will want to make sure that the query includes all the columns containing values that you want to use to populate new records. Any fields that are not mapped will not be populated in the new records. The process of defining mappings is facilitated by the **Append To** cell, which appears in the design grid when you create an Append query and allows you to select the target field to which each source field corresponds.

Only Super Users and members of the MI Power User Role Security Group can create Append queries. All users can *run* Append queries, provided that they have been granted permission to access the Catalog folder in which they are stored. To create the records returned by the query, a user must have Insert privileges on the family in which the records are being created.

Example of a Crosstab Query

When management personnel request that work be performed on piece of equipment, the work results in some amount of downtime for the piece of equipment. Depending on the type of work being performed, some activities result in longer amounts of downtime than others. In your company, work requests are recorded in Work Request records, which contain the following fields:

- **Work Activity:** Indicates the type of work that should be performed (e.g., repair).
- **Work Request ID:** Identifies the work request with a unique value.
- **Downtime:** Indicates the total amount of time that the equipment was out of service while work was being performed on it.

If you were to create a Select query to view information about work requests that have been completed, the results might look something like those shown in the following image.

Work Request ID	Work Activity	Downtime
WR1	Repair	1
WR2	Adjust	2
WR3	Clean	1
WR4	Repair	8
WR5	Repair	2
WR6	Adjust	1
WR7	Clean	4
WR8	Repair	9
WR9	Adjust	1
WR10	Clean	1

From these results, you can see each work request ID, the corresponding work activity, and the total amount of downtime per request. The format of these results, however, does not display the work requests grouped by activity type. While this result set is small, which allows you to visually determine how many work requests fall into each activity type, more typical query results will contain enough rows of data that it will be difficult to divide it into categories by visually comparing the data. This is especially true when the results span multiple pages.

To group the results so that you can see at a glance how many work requests fall into each activity type, you could decide to make the query an aggregate query. If you use the COUNT function on the Work Request ID field and the SUM function on the Downtime field, the query results would look similar to those shown in the following image.

Work Request ID	Work Activity	Downtime
3	Adjust	4
3	Clean	6
4	Repair	20

From these results, you can see that three work requests asked for an adjustment, three requests asked for something to be cleaned, and four work requests asked for a repair. You can also see that the total amount of downtime for all adjustments was four days, the total amount of downtime for all cleaning tasks was six days, and total amount of downtime for all repairs was 20 hours.

While the stored data is interesting when viewed in this format, you might be more interested in determining which work requests resulted in a downtime *over* a certain number of days. For instance, suppose that you expect repairs to take over seven days, but cleaning tasks that take more than seven days are unacceptable to management personnel. You might want to construct the query such that it groups the raw data into two categories: downtime and work activity. You then want to determine how many work requests in each type of activity resulted in downtime between one and seven days and how many resulted in downtime over seven days.

The results of a Select query cannot present the data in this format. To format the data such that it provides the desired information, you must create a Crosstab query, where you can:

- Convert the stored downtime values into categories: *1 to 7 Days* and *Over 7 Days*.
- Determine the total amount of downtime per work activity.
- Divide the total amount of downtime per work activity into the pre-defined categories of *1 to 7 Days* and *Over 7 Days*.

The Crosstab query will contain the same fields as the Select query: Work Activity, Work Request ID, and Downtime.

To convert the stored downtime values into categories, however, you will need to add another column that includes a DECODE statement that uses the SIGN function.

The DECODE statement would look like this:

Decode(SIGN(([Work Request].[Downtime] - 7)), -1, '1-7 Days', 0, '1-7 Days', 'Over 7 Days')

This statement indicates that:

- First, the value seven should be subtracted from the actual downtime values. Because the Downtime field has a unit of measure of Days, and there are seven days in a week, each downtime value will be greater than or equal to 0 and less than or equal to 7. After subtracting the value seven from these stored downtime values, the calculated result will be either a negative number, zero, or a positive number.
- After subtracting seven from the actual downtime value:
  - If the value is negative or zero (0), the record should be grouped into the category *1-7 Days*. This means that any work request with the following downtime values will be grouped into the *1-7 Days* category: 0, 1, 2, 3, 4, 5, 6.
  - If the value is anything other than zero or a negative number, the record should be grouped into the category *Over 7 Days*. This means that any work

request with a downtime value greater than or equal to eight (8) will be grouped into the category *Over 7 Days*.

In the query design grid, the Work Activity field will be the row header, and the column with the DECODE statement will be the column header. The Work Request ID field will be the intersecting field, or the **Value**, and a COUNT function will be defined in the **Total** cell for the Work Request ID field. This means that the intersecting cell in the results will contain a number instead of a Work Request ID. The number will indicate the number of work requests that fall into the category defined by the intersection of the row and the column (e.g., the number of repair work requests that resulted in a downtime of over seven days).

In addition, the SUM function will be defined in the **Total** cell for the Downtime field. This will ensure that the results contain only one row representing each work activity instead of multiple rows containing the same work activity. For example, if there are three repair requests, because the SUM function is defined on the Downtime field, the results will contain only one row representing the repair work type (displaying the *total* number of work requests of that type) instead of three rows representing the repair work type (displaying only *one* work request of that type for each row).

The results will be grouped the way they are shown in the following table.

	Downtime Category (1-7 Days)	Downtime Category (Over 7 Days)
Work Activity (Repair)	# of Repairs with a Downtime of 1-7 Days	# of Repairs with a Downtime Over 7 Days
Work Activity (Clean)	# of Cleaning Tasks with a Downtime of 1-7 Days	# of Cleaning Tasks with a Downtime Over 7 Days
Work Activity (Adjust)	# of Adjustments with a Downtime of 1-7 Days	# of Adjustments with a Downtime Over 7 Days

The design grid would look something like the following image.

[Work Request]

[Downtime]
[Work Activity]
[Work Request ID]
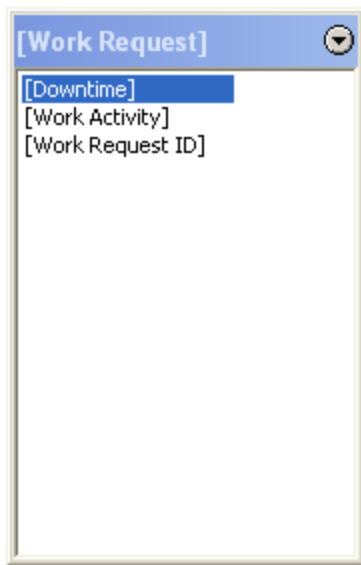
Field	[Work Activity]	Decode(SIGN(([Work...	[Work Request ID]	[Downtime]
Alias	Work Activity			Downtime
Table	[Work Request]		[Work Request]	[Work Request]
Total	Group By	Group By	Count	Sum
Crosstab	Row Heading	Column Heading	Value	
Sort				
Sort Index				
Criteria				
Or				

The results of the query would look something like those shown in the following image.

Work Activity	1-7 Days	Over 7 Days
Adjust	3	0
Clean	3	0
Repair	2	2

Overview of Reports

A report is a formatted view of data that exists in the Meridium APM database. Reports provide you with a way of formatting information for display, export, and printing purposes.

You can create a report from query results, dataset results, or by importing a report that was saved to an external location. After you create a report, you can save it so that you or others can run it repeatedly. Like other items, reports are saved in the Catalog and can be accessed via hyperlinks on the Home Page.

To create a report, you will use **SQL Server Reporting Services**. If you want to create a report within the Meridium APM Framework application, Microsoft SQL Server Business Intelligence Development Studio must be installed on the Meridium APM client machine. In addition, a SQL Server Report Server must be configured, and that server must be referenced via the Configuration Manager using the **SQL Server Report Server Name** setting.

After a report has been saved to the Catalog, you can view that report using the **Report Viewer** page. Note that you will need the following permissions to view reports:

- View/Execute permissions on the Catalog folder in which the report is located.
- At least View privileges on all families included in the underlying query or dataset.

Overview of R Scripts

R Scripts expand the functionality of the Meridium APM Framework application, and Policies. R Scripts in the Meridium APM Framework application provide a way for you to utilize the R programming language to create scripts that can receive data from inputs, make calculations, and then return one or more outputs. R Scripts are used with Policy Manager.

You can create an R Script by writing or pasting R code into the **R Script Editor**. After you create and save an R Script, you can use R-Script nodes in Policy Manager to pass values in to the R Script, and receive calculated values out of the R Script.

To create an R Script, you will use the **R Script Editor**. If you want to create R Scripts within the Meridium APM Framework application, Revolution Analytics must be installed on the Meridium APM Application Server.

Overview of Strategy Rules

Strategy rules exist for the purpose of defining actions that you want the Meridium APM system to perform. These rules are defined in VSTA. When you create and save a Strategy record, the attributes of the Strategy record are saved, an associated Rules Library rule project is created, but no rules are created within the rule project. You will need to create strategy rules manually and compile the strategy rule code. Strategy rules specify the criteria that the Meridium APM system will use to carry out a particular action, such as generating a list of Task records within a given Task family.

Strategy rules are classified as one of two types:

- Base strategy rules.
- Sub-strategy rules that are called by base strategy rules.

When you create strategy rules, you can specify that the rules should perform some action based on query or dataset results. You can specify that you want the rules to:

- Create a Task record.
- Create an Alert record, which will send an email notification to the desired recipient(s).
- Invoke another strategy rule.

Meridium APM uses a tool called the Meridium APM Rules Editor to display strategy rules. The Meridium APM Rules Editor allows you to create basic logical rules that support all basic logical and arithmetic operators (e.g., AND, OR, NOT, +, -, /). Before you can begin writing strategy rules, you must first have an understanding of Visual Studio Tools for Applications (VSTA).

Strategy rules are stored as projects in the Meridium APM Rules Library in the folder \\Client\Strategy Rules. When you create a new Strategy record, a strategy rule project is created automatically and saved in the Rules Library folder \\Client\Strategy Rules. The name of the rule project corresponds to the ID of the Strategy record. The rule project is created using a template that serves as a staring point for writing strategy rule logic. You must understand how to use VSTA in order to modify the template and write strategy rules.

> Note: Baseline strategy rules are stored in the Rules Library in the folder \\Meridium\Strategy Rules and cannot be modified.

Because strategy rules are stored in the Rules Library, to manage strategy rules *fully*, you must be able to access the Rules Library in the Configuration Manager application. In the Meridium APM Framework application, you will be able to access, modify, and compile the rules. You will not, however, have access to any other Rules Library content or functions. For more information on the Rules Library, see the Configuration Manager Help.

After strategy rules have been created, you can schedule them for execution or execute them manually. The strategy rules will not cause the Meridium APM system to perform any actions until the rules are executed.

Baseline Strategy Rules

The following strategy rules are provided as part of baseline Meridium APM content:

- **MI_ProcessConditionalAlerts:** When executed, searches the database for all Conditional Alerts records. If any records exist, the strategy rule uses the information in those records (i.e., interval at which to send the email message and the date the email message was last sent) to determine if an email message should be sent. If an email message should be sent, an Alert record is created and the email message is sent.

- **MI_InspectionTasksUpdate:** When executed, updates the values in the Last Date and Next Date fields in all Inspection Task records that are linked to Inspection Event records that have a value in the Completion Date field. Note, however, that the value in the Inspection Task Complete field must be set to *True* in those Inspection Event records in order for this to work correctly. If the value in the Inspection Task Complete field is set to *False*, the Inspection Task records to which the Inspection Event record is linked will not be modified.

  For example, suppose an Inspection Event record is linked to an Inspection Task record, and the Inspection Event record contains the value *May 1, 2006* in the Completion Date field. If you execute the MI_InspectionTasksUpdate strategy rule and the Inspection Task Complete field is set to *True*, the Last Date in the Task record will be updated to *May 1, 2006*, and the Next Date will be updated according to the desired interval in that Inspection Task record.

  > **Note:** In the baseline Meridium APM database, Inspection Management is configured to use this strategy rule. If desired, you can [create your own custom strategy rule](#) and specify that you want to use it instead to update dates in Inspection Task records.

About Sub-strategy Rules

Sub-strategy rules are strategy rules that are *always* called by other strategy rules. The Meridium APM system cannot directly execute sub-strategy rules. If you do not select the **Sub-Strategy** check box on the **Create Strategy** screen when you create and save a new Strategy record, the associated strategy rules will be implemented as base strategy rules by default.

Usually, sub-strategy rules will perform *reusable* work. Note that many base strategy rules can call the same sub-strategy rules to make use of already existing functionality. Sub-strategy rules can also call other sub-strategy rules.

Note that strategy rules are not *required* to call sub-strategy rules. Strategy rules can be *independent*, meaning that the Meridium APM system can perform an action by simply executing those strategy rules without calling other sub-strategy rules.

Sub-strategy rules that are created for one family cannot be called by base strategy rules that were created for a different family.

> **Note:** Sub-strategy rules can be defined only upon creation of the associated Strategy record. The sub-strategy rule designation becomes a permanent setting after you save the Strategy record.

About Strategy Records

While Strategy records are just records in a family, you will not usually view them in the Record Manager or Bulk Data Form like you would other Meridium APM records. Instead, you will probably find it useful to think of a Strategy record as an organizational object whose purpose is to store rules that tell the Meridium APM system what to do, how to do it, and when to do it.

Strategy records and strategy rules are closely connected. While a Strategy record can exist without any associated strategy rules, any Strategy record that does not have any strategy rules will not do anything.

For example, the baseline Strategy record MI_InspectionTasksUpdate, which is associated with the Inspection family, has a strategy rule that causes Inspection Task records to be updated when an Inspection record's Completion Date field is populated and the record is saved. The rule dictates, along with other specifications, that when an Inspection record is assigned a completion date, the value in the Completion Date field of the Inspection record will be used to populate the Last Date field of the Task record listed in the **Tasks Addressed** cell of the Inspection record. The rules tell the Meridium APM system how to use the Strategy record. Without them, no actions would be performed.

When you create a Strategy record, you must associate it with an entity family. You can then define various information for the Strategy record in fields like you can for other records. Strategy records belong to the Strategy family, which means that you can search for Strategy records by searching the Strategy family.

Strategy records have attributes that describe the purpose, basis, and corporate guideline for the associated strategy rules. For example, the purpose of the MI_InspectionTasksUpdate strategy rules, as indicated in the Purpose field of the Strategy record, is *to update inspection tasks when documenting inspections*. In addition, you can specify industry standards that apply to the associated strategy rules.

Accessing Strategy Rules

To access Strategy Rules:

- In the Meridium APM Framework, on the main menu, click **Tools**, and then click **Strategy Rules**.

  The **Strategy Start Page** appears.

Aspects of the Strategy Start Page

The **Strategy Start Page** contains the following links:

- **Create Strategy:** Lets you choose an entity family and [create a Strategy record associated with that family](#). Only members of the MI SM Administrator Security Group can perform this task. If you are not a member of the MI SM Administrator Security Group, when you click this link, a message appears, indicating that you do not have sufficient permission.

- **View Strategies:** Displays the [**Strategy List** page](#), where you can view all Strategy records that exist in the Meridium APM database. Only members of the MI SM Administrator Security Group can perform this task. If you are not a member of the MI SM Administrator Security Group, when you click this link, a message appears, indicating that you do not have sufficient permission.

Overview of Task Management

Task Management allows you to manage all Task, Task Revision, and Task Execution records that exist in the database.

- A *Task* record stores information about an action that should be performed.
- A *Task Revision* record captures changes to values in Task records. Each time you modify a value in a Task record, a new Task Revision record is created to capture the information in the Task record as it existed on that date and time.
- A *Task Execution* record captures the date and time at which the action was actually performed.

Because State Configuration has been defined for the Task family, when a Task record is created, it is assigned a Proposed state. From the Proposed state, you can transition the Task record to a variety of other states, such as Scheduled or Rejected.

In addition, Task Management allows you to view a graphical representation of Task records using the Timeline.

Task Management Data Model

The following diagram illustrates how families are related to one another within the Task Management data model.
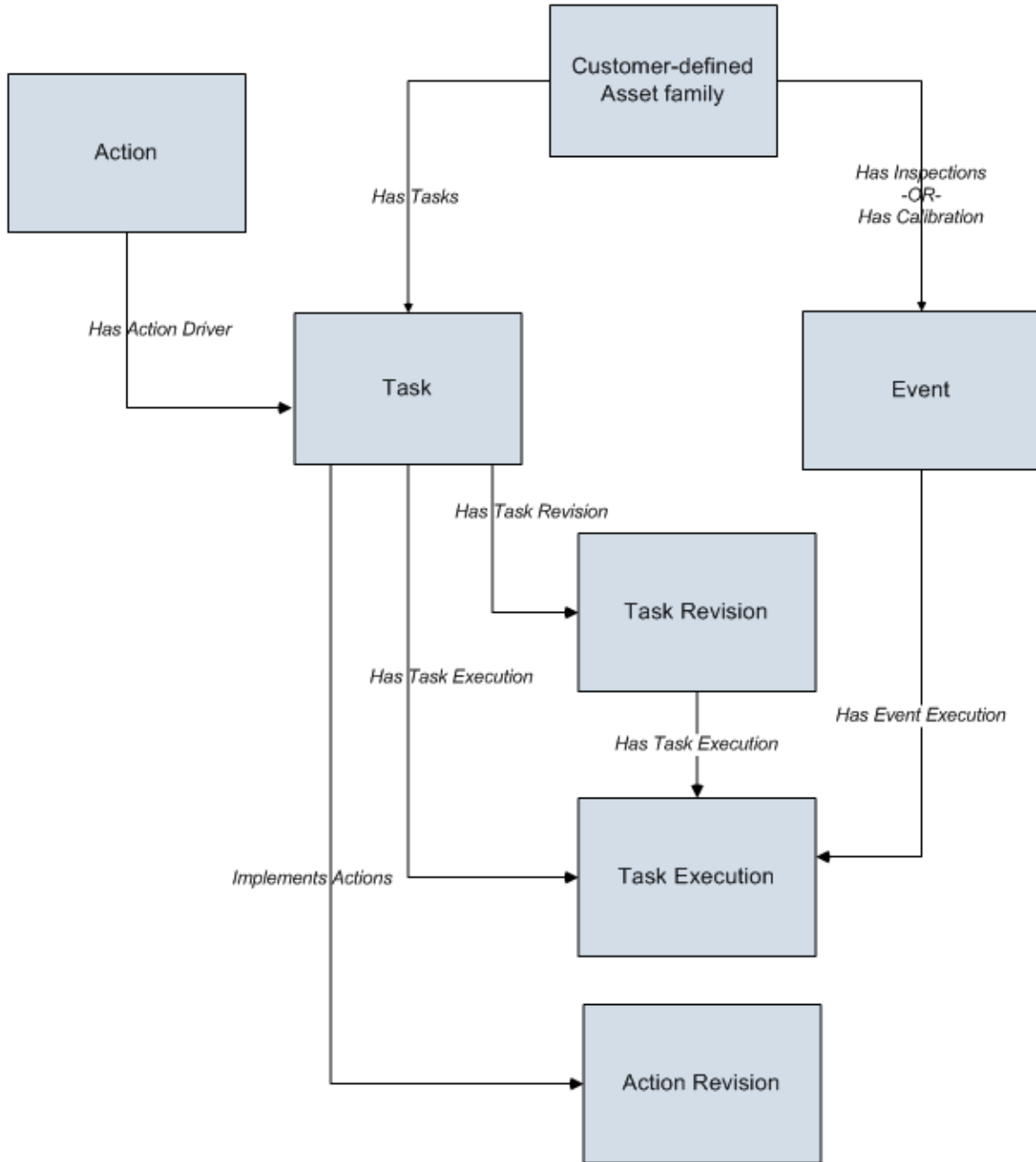
How State Configurations are Used in Task Management

Task Management employs standard State Configuration functionality to facilitate basic Task Management operations and record state transitions. In Task Management, the **Selected Tasks** menu displays a list of operations that are available for the record(s) that is currently selected. The list of available operations will vary, depending upon the current state of the selected record and the states to which the record can transition from its current state, according to the State Configuration that exists for the entity family.

For example, in the baseline Meridium APM database, a Task record is assigned an initial state of Proposed when it is first added to the database. From the Proposed state, a Task record can be moved to the Rejected state via the Reject operation or the Scheduled state via the Schedule operation. Therefore, when you select a Task record that is in the Proposed state, the **Selected Tasks** menu will display two links: **Reject** and **Schedule**. When you click the **Reject** link, the record will transition from the Proposed state to the Rejected state.

With the exception of reserved states, the links that appear on the **Selected Tasks** menu are the same as the options that would appear on **Operations** menu in the datasheet in Record Manager, provided that the State Configuration has been set up to show operations on datasheets. The **Selected Tasks** menu simply provides a custom view of the standard State Configuration functionality, which is also available in the Record Manager.

The **Selected Tasks** menu will display ANY operation that is valid for the current state of the selected record, including any baseline operations and any custom operations. Note also that the **Selected Tasks** menu displays reserved operations that are not available in the Record Manager. Throughout this documentation, we provide details on performing baseline operations for Task families using their baseline State Configurations. If you have customized the State Configurations for the Task families, additional options may be available to you.

Task States and Operations

The following diagram shows the states and operations that exist in the baseline State Configuration for the Task family. You can use this diagram to determine which operations will appear on the **Selected Tasks** menu when you select a Task record in the grid on the **Task List** page. The **Selected Tasks** menu will display any operation *to which* the current record can be transitioned from its current state.

> **Note:** In the baseline database, State Configuration is defined for the root Task family and spread to all subfamilies. State Configuration is enabled, however, only for the root Task family. If desired, you can enable State Configuration for other baseline subfamilies (e.g., Inspection Task). States and Operations

States and Operations

The following State Configuration is defined for the Task family. Note that in this image, a box represents a state, and an arrow represents an operation. The state to which an arrow points represents a successor state.

Proposed

Rework · Schedule

Scheduled

Re-Schedule · Execute

Executing

Mark as Complete

Reject

Reject

Mark as Complete

Completed

Rejected
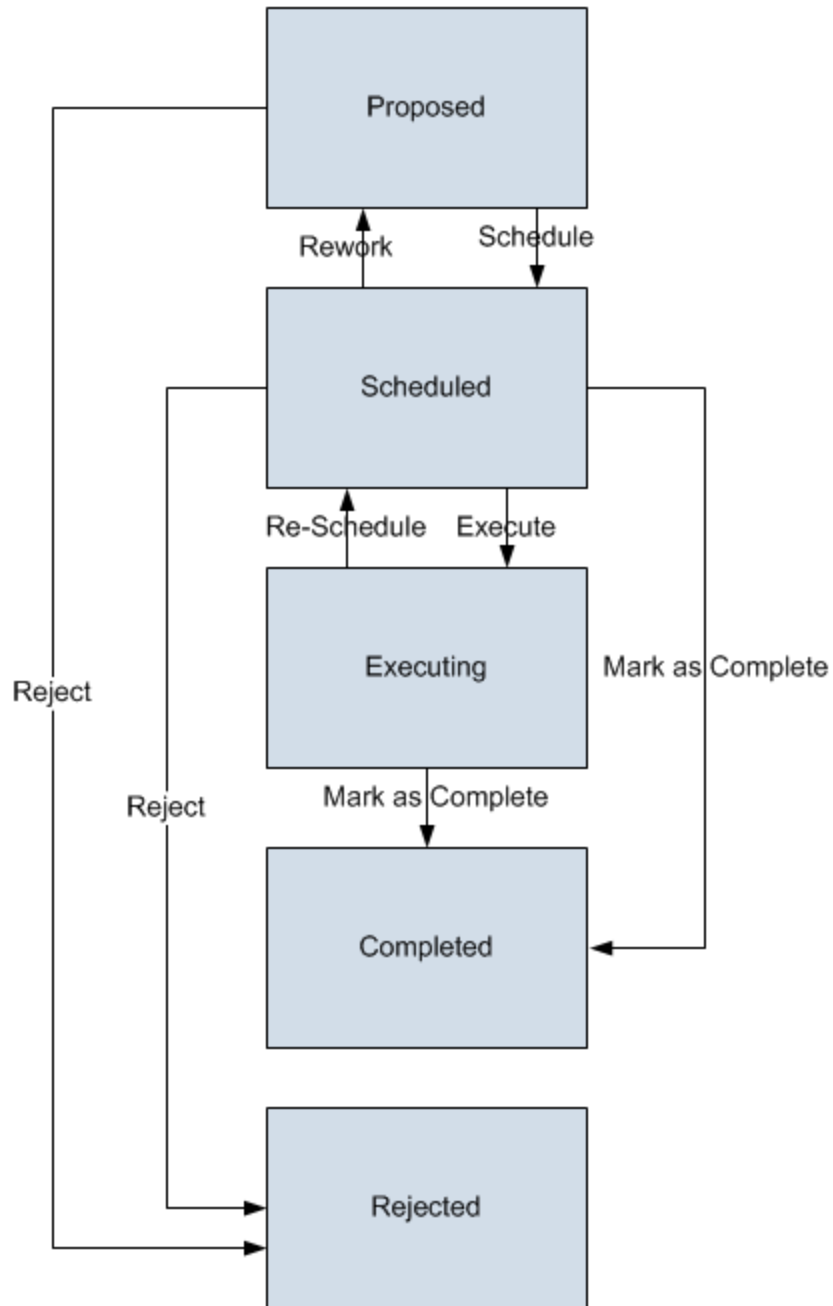
Initial State

The Proposed state is the initial state of all new Task records.

Datasheet Configuration

By default, states and operations will appear on the datasheet when you are viewing a Task record in the Record Manager or the Bulk Data Form.

Reserved States and Operations

The following table lists the baseline states and operations and indicates which of these states and operations are reserved. You cannot remove or modify reserved states or operation. You can, however, add your own states and operations to the State Configuration.

States	
State	**Is Reserved?**
Completed	Yes
Executing	Yes
Proposed	No
Rejected	Yes
Scheduled	Yes
Operations	
Operation	**Is Reserved?**
Mark as Complete (where Executing is the predecessor state)	Yes
Re-Schedule	Yes
Reject (where Proposed is the predecessor state)	Yes
Schedule	Yes
Execute	Yes
Mark As Complete (where Scheduled is the predecessor state)	No
Reject (where Scheduled is the predecessor state)	Yes
Rework	No

State Configuration Roles

By default, NO State Configuration Roles are assigned to any state in the Task State Configuration. If desired, you can assign State Configuration Roles to any state. In addition, for each state, the **Require a specific user to be assigned to a state** check box is *selected*.

Accessing the Task Management Functions Page

To access the Task Management Functions page:

- In the Meridium APM Framework, on the main menu, click **Tools**, and then click **Tasks**.

  The [Task Management Functions page](#) appears.

  > **Note:** Throughout the documentation, we refer to this page as the *Task Management Start Page*.
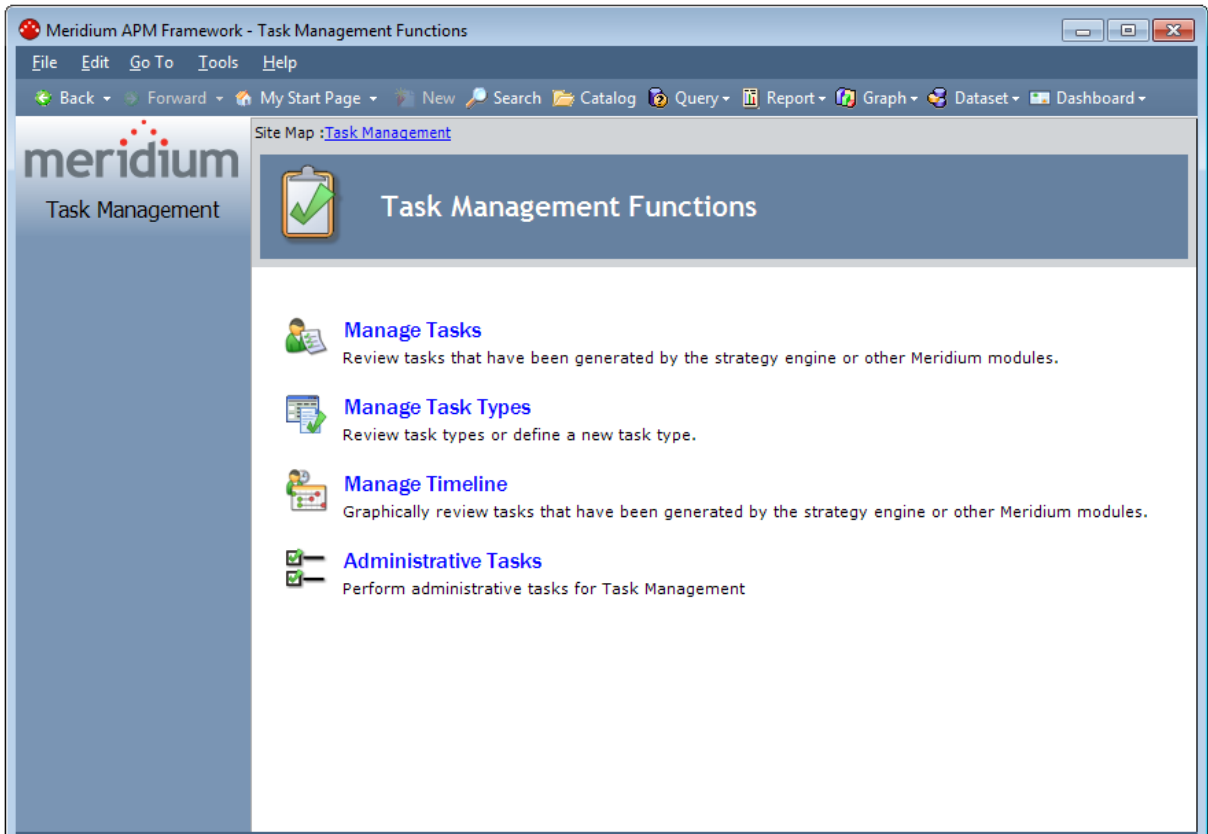
Aspects of the Task Management Functions Page

The **Task Management Functions** page contains the following links:

- **Manage Tasks:** Displays the **Task List** page, where you can create new Task records, open existing Task records, or change the state of an existing Task record. You can also view Task Revision, Task Execution, and Recommendation records.

- **Manage Task Types:** Displays the **Find Items** window, where you can select an existing Task Type record that you want to view.

- **Manage Timeline:** Displays the **Task Query Builder**, where you can select Task records to compare visually against global event records on the Timeline.

- **Administrative Tasks:** Displays the **Task Management Administration** page, where you can specify a Recommendation Query and a Task Query.

> **Note:** The **Administrative Tasks** link is available only if you are logged in as a Super User or a member of the MI Task Manager Administrator Security Group.

First-Time Deployment Workflow

Deploying and configuring the Core Analysis features for the first time includes completing multiple steps, which are outlined in the table in this topic. The steps in this section of the documentation provide all the information that you need to deploy the Core Analysis features within the basic Meridium APM system architecture.

The Core Analysis feature that each step supports is indicated in the **Notes** column. Whether a step is required or optional is indicated in the **Required/Optional** cell. Steps are marked as *Required* if you must perform the step to take advantage of the Core Analysis functionality.

The person responsible for completing each task may vary within your organization. We recommend, however, that each step be performed in relatively the same order in which they are listed in the following table.

Step	Task	Required/Optional	Notes
1	Assign Security Users to the MI SM Administrator Security Group.	Required	Supports Strategy Rules.
2	Assign Security Users to the Task Management Security Groups.	Required	Supports Task Management.
3	Configure additional family-level privileges needed for using strategy rules.	Required	Supports Strategy Rules.
4	Configure Meridium APM to Create Task Revision records.	Optional	Supports Task Management. This step is necessary only if you want the Meridium APM system to create Task Revision records automatically.

Step	Task	Required/Optional	Notes
5	Filter the Task Types List in Task records.	Optional	Supports Task Management This step is necessary only if you want to filter the Task Types list in Task records.
6	Configure Catalog security permissions for users who will be working with dashboards.	Required	Supports Dashboards.
7	Configure Catalog security permissions for users who will be working with strategy rules.	Required	Supports Strategy Rules.
8	Identify Equipment and Functional Location records.	Optional	Supports Task Management. This step is necessary only if you use families *other than* the baseline Equipment and Functional Location families to store data about the equipment or location, respectively.
9	Define the Recommendation query.	Optional	Supports Task Management. This step is necessary only if you do *not* want to use the baseline Recommendation query.

Step	Task	Required/Optional	Notes
10	Define the Task query.	Optional	Supports Task Management. This step is necessary only if you do *not* want to use the baseline Task query.
11	Install the third-party software that supports R Scripts.	Required	Supports R Scripts. R Scripts in the Meridium APM Framework application will not function without the third-party software. You must install this software according to the system requirements.
12	Configure your R Script credentials.	Required	Supports R Scripts. You must specify the credentials that will be used to connect to the third-party software that supports R Scripts.

Upgrade or Update Core Analysis to 3.6.1.3.0

The following table lists the step that is required to update or upgrade Core Analysis to 3.6.1.3.0. This step assumes that you have completed the steps for upgrading the basic Meridium APM system architecture.

Update from any version V3.6.1.0.0 through V3.6.1.2.0

This module will be updated to 3.6.1.3.0 automatically when you update the components in the basic Meridium APM system architecture. No additional steps are required.

Upgrade from any version V3.6.0.0.0 through V3.6.0.12.6

Step	Task	Location	Required/Optional	Notes
1	Specify the query privilege setting for your Security Users.	Configuration Manager	Optional	The query privilege setting allows you to set restrictions on the tasks that Security Users can perform with queries. Note that if you do not specify a query privilege setting for your Security Users, query behavior will remain the same for all Security Users. In other words, unless you want to take advantage of the additional level of security for queries, you do not have to perform this step.

Step	Task	Location	Required/Optional	Notes
2	Install the third-party software that supports R Scripts.		Required	Supports R Scripts. R Scripts in the Meridium APM Framework application will not function without the third-party software. You must install this software according to the system requirements.
3	Configure your R Script credentials.	Configuration Manager	Required	Supports R Scripts. You must specify the credentials that will be used to connect to the third-party software that supports R Scripts.

Upgrade from any version V3.5.1.0.0 through V3.5.1.12.0

Step	Task	Location	Required/Optional	Notes
1	Specify the query privilege setting for your Security Users.	Configuration Manager	Optional	The query privilege setting allows you to set restrictions on the tasks that Security Users can perform with queries. Note that if you do not specify a query privilege setting for your Security Users, query behavior will remain the same for all Security Users. In other words, unless you want to take advantage of the additional level of security for queries, you do not have to perform this step.
2	Install the third-party software that supports R Scripts.		Required	Supports R Scripts. R Scripts in the Meridium APM Framework application will not function without the third-party software. You must install this software according to the system requirements.

Step	Task	Location	Required/Optional	Notes
3	Configure your R Script credentials.	Configuration Manager	Required	Supports R Scripts. You must specify the credentials that will be used to connect to the third-party software that supports R Scripts.

Upgrade from any version V3.5.0 SP1 LP through V3.5.0.1.9.0

Step	Task	Location	Required/Optional	Notes
1	Specify the query privilege setting for your Security Users.	Configuration Manager	Optional	The query privilege setting allows you to set restrictions on the tasks that Security Users can perform with queries. Note that if you do not specify a query privilege setting for your Security Users, query behavior will remain the same for all Security Users. In other words, unless you want to take advantage of the additional level of security for queries, you do not have to perform this step.

Step	Task	Location	Required/Optional	Notes
2	Install the third-party software that supports R Scripts.		Required	Supports R Scripts. R Scripts in the Meridium APM Framework application will not function without the third-party software. You must install this software according to the system requirements.
3	Configure your R Script credentials.	Configuration Manager	Required	Supports R Scripts. You must specify the credentials that will be used to connect to the third-party software that supports R Scripts.

Upgrade from any version V3.5.0 through V3.5.0.0.7.1

Step	Task	Location	Required/Optional	Notes
1	Specify the query privilege setting for your Security Users.	Configuration Manager	Optional	The query privilege setting allows you to set restrictions on the tasks that Security Users can perform with queries. Note that if you do not specify a query privilege setting for your Security Users, query behavior will remain the same for all Security Users. In other words, unless you want to take advantage of the additional level of security for queries, you do not have to perform this step.
2	Install the third-party software that supports R Scripts.		Required	Supports R Scripts. R Scripts in the Meridium APM Framework application will not function without the third-party software. You must install this software according to the system requirements.

Step	Task	Location	Required/Optional	Notes
3	Configure your R Script credentials.	Configuration Manager	Required	Supports R Scripts. You must specify the credentials that will be used to connect to the third-party software that supports R Scripts.

Upgrade from any version V3.4.5 through V3.4.5.0.1.4

Step	Task	Location	Required/Optional	Notes
1	Specify the query privilege setting for your Security Users.	Configuration Manager	Optional	The query privilege setting allows you to set restrictions on the tasks that Security Users can perform with queries. Note that if you do not specify a query privilege setting for your Security Users, query behavior will remain the same for all Security Users. In other words, unless you want to take advantage of the additional level of security for queries, you do not have to perform this step.

Step	Task	Location	Required/Optional	Notes
2	Install the third-party software that supports R Scripts.		Required	Supports R Scripts. R Scripts in the Meridium APM Framework applic-ation will not func-tion without the third-party software. You must install this software according to the system requirements.
3	Configure your R Script credentials.	Configuration Manager	Required	Supports R Scripts. You must specify the credentials that will be used to con-nect to the third-party software that supports R Scripts.

Dashboard Security

Security permissions to the Catalog folders where dashboards and their underlying queries are stored determine how users can interact with dashboards in the Meridium APM Framework. Because all Security Users have inherent execute permissions to the Public Catalog folder and subfolders *and* because the module-specific Security Groups provide the necessary permissions to the families that are used in the underlying queries to baseline dashboards, all users that are assigned to the appropriate module-level Security Groups can [work with these dashboards](#). *Additional* security permissions are necessary, however, to perform some operations related to dashboards.

The following table lists the specific dashboard operations that require security permissions beyond the inherent execute permissions that are granted to all users by default. It also identifies the specific permissions that are needed to perform those operations.

Operation	Security permission
Create a new dashboard	Create
Delete a dashboard	Delete
Add a section to a dashboard	Edit
Remove a section from a dashboard	Edit
Modify the content of a dashboard section	Edit

MI SM Administrator Security Group

The MI SM Administrator Security Group is provided in the baseline Strategy Rules tool. Users who are members of this Security Group can:

- Create new Strategy records by clicking the **Create Strategy** link on the **Strategy Start Page**.
- View a list of existing Strategy records by clicking the **View Strategies** link on the **Strategy Start Page**.
- Manage strategy rules.
- Execute strategy rules.

Users who are not members of this Security Group will not be able to perform any of these tasks. This means that if a user who is not a member of this Security Group clicks the **Create Strategy** or **View Strategies** link on the **Strategy Start Page** in the Meridium APM Framework application, that user will receive a message indicating that he or she does not have appropriate privileges.

> **Note:** If a user is not a member of the MI SM Administrator Security Group but they DO have Insert privileges to the Strategy family, that user can create a Strategy record by clicking the **New** button on the Meridium APM Framework toolbar. This method of creating Strategy records is not recommended, however. Instead, we recommend that users create Strategy records via the **Strategy Start Page**, which means that you should add users to the MI SM Administrator Security Group if you want them to be able to do this.

In addition to assigning Security Users to the MI SM Administrator Security Group, you must assign the appropriate family-level privileges to any Security User or Security Group who should be able to perform tasks related to strategy rules or Strategy records. Specifically, you must set family-level privileges for any customer-defined families that will be used to create strategy rules.

> **Note:** Super Users do not need to belong to the MI SM Administrator Security Group or have family privileges to perform tasks in the Strategy Rules tool. Being a Super User alone allows you to perform all tasks.

Family-Level Privileges Needed by Strategy Rules Users

Users who are not members of the MI SM Administrator Security Group cannot access any functionality via the **Strategy Start Page**. Being a member of this Security Group, however, is not sufficient for performing all the tasks available in the Strategy Rules tool.

In addition to being a member of the MI SM Administrator Security Group, users need certain family-level privileges. The following table describes the tasks that can be performed and the family-level privileges needed to accomplish each task.

To perform this task...	You need these family privileges...
Create a Strategy record associated with <Family A>.	View privileges on <Family A>. Insert privileges to the Strategy family.
Open a Strategy record associated with <Family A>.	View privileges to the Strategy family.
Execute a strategy rule that references a query.	View privileges to any family included in the query. Note that you also need View/Execute privileges to the Catalog folder (and any folders of that Catalog folder) in which the query exists.
Execute a strategy rule that creates a new record.	Insert privileges to the family of the new record.
Execute a strategy rule that updates an existing record.	Update privileges to the family of the existing record.
Execute a strategy rule that references a sub-strategy rule.	The privileges you will need depend on the code defined in the sub-strategy rule.

Note that if a strategy rule references multiple Meridium APM items, users will need the family privileges for *each* item. So, if a strategy rule creates a Task record and an Alert record, to execute the strategy rule, a user would need Insert privileges to the Task family and the Alert family.

> **Note:** Before users can create Strategy record, they will need View privileges to the following Catalog folder: \\Public\Rules Library\Meridium\Strategy Rules. They will also need Insert privileges to the following Catalog folder: \\Public\Rules Library\Client\Strategy Rules.

Configuring Catalog Security Permissions for Strategy Rules Users

The following table lists the permissions that users will need for creating Strategy records.

Catalog Folder	Permissions Required
\\Public\Rules Library\Client\Strategy Rules	Create
\\Public\Rules Library\Meridium\Strategy Rules	View/Execute

The following table lists the permissions that users will need for deleting Strategy records.

Catalog Folder	Permissions Required
\\Public\Rules Library\Client\Strategy Rules	Delete

Note the following points:

- The Catalog folder \\Public\Rules Library\Client\Strategy Rules is not provided in the baseline Meridium APM database. This folder is created automatically after a new Strategy record is saved. If you want non-Super Users to be able to create new Strategy records, you will need to create this folder manually before they attempt to create a new Strategy record. You can create the folder manually by:

    - Creating a foldervia the Meridium APM Framework application.

    - Adding the folder to the Rules Library via the Configuration Manager

    - Logging in as a Super User and creating a temporary Strategy record (Super Users do not need any explicit Catalog permissions to be able to create Strategies) that you will later delete.

- If you want to grant permissions to a subfolder, the user must also have permissions on the main folder. For example, a user who has View permissions to the Catalog folder \\Public\Rules Library\Client will be able to view that folder only if View permissions have also been granted to the Catalog folders \\Public\Rules Library and \\Public. To streamline the process, you can grant permissions to a folder and spread the permissions to all subfolders of that root folder.

    Note: Spreading permissions from a folder will not spread the permissions to subfolders that are not yet created. This means that even if you grant a user Create permissions to the Catalog folder \\Public\Rules Library\Client, that user will have automatic Create permissions to the Catalog folder \\Public\Rules

Library\Client\Strategy Rules *only* if the subfolder exists at the time you grant the permissions to the root folder.

- All Security Users automatically have inherent *execute* permissions for all Public Catalog folders. This means that for a user to be able to execute strategy rules from the **Strategy List** page, they only need to be a member of the MI SM Administrator Security Group. They do not need explicit Catalog permissions.

    Note: Users that are members of the MI SM Administrator Security Group can execute strategy rules only if the baseline MI SM Administrator family-level privileges have not been modified. If the privileges have been modified, the user must be a member of the MI SM Administrator Security Group and have at least View privileges on the Strategy family.

- All strategy rules users must be a member of the MI SM Administrator Security Group to create or delete Strategy records.
- All strategy rules users must have the appropriate permissions to the Strategy family to manage strategy rules. Other family-level permissions might also be required for users to manage strategy rules.

Task Management Security Groups

Meridium APM provides the following baseline Security Groups for use with Task Management:

- MI Task Manager Administrator
- MI Task Manager User

The following table lists the family-level privileges that exist for these Security Groups.

Entity Families		
Family	MI Task Manager Administrator	MI Task Manager User
Action	None	None
Action Revision	None	View
Equipment	View	View
Event	None	None
Functional Location	View	View
Measurement Location Group	View	None
RCM FMEA Asset	View	View
SAP System	None	View
Task	View, Update, Insert, Delete	View, Update, Insert
Task Execution	View, Update, Insert, Delete	View, Update, Insert
Task Types	View, Update, Insert, Delete	View, Update, Insert
Relationship Families		
Family	MI Task Manager Administrator	MI Task Manager User
Has Action Driver	None	None
Has Event Execution	View, Update, Insert, Delete	View, Update, Insert
Implements Action	None	View

Entity Families		
Has Tasks	View, Update, Insert, Delete	View, Update, Insert
Has Task Execution	View, Update, Insert, Delete	View, Update, Insert
Has Task History	None	None
Has Task Revision	View, Update, Insert, Delete	View, Update, Insert

About Configuring Meridium APM to Create Task Revision Records

If desired, you can configure Meridium APM to create records that track changes to Task record values. Doing so allows you to keep an historical record of Task data on a given date and time. Throughout this documentation, we refer to these revision-tracking records as *Task Revision records*. Keep in mind, however, that the family caption is not necessarily Task Revision.

Meridium APM provides the following Task Revision families, but you might want to create your own:

- Task Revision
- Calibration Task Revision
- Inspection Task Revision

If you configure Meridium APM to create Task Revision records, when you create a Task record, Meridium APM will automatically create a Task Revision record and link it to the Task record. If you later update the Task record, a new Task Revision record will be created.

The Meridium APM system assumes that you do not want a Task Revision record to be created when you create a new Task record or update an existing Task record. If, however, you decide that you want these Task Revision records to be created, you will need to perform the following steps:

1. Identify the family of the Task records whose revisions you want to track. The family must be the root Task family or one of its subfamilies (either a baseline subfamily or a customer-defined subfamily).

2. Make sure that the selected family has a subfamily that will store the Task Revision records. If it does not, add a Task Revision subfamily to the selected family.

3. Configure the Has Task Revision relationship to include the selected Task family as the predecessor and the Task Revision family as the successor.

4. If the Task Revision family that you identified in step 2 is not a baseline Task Revision family, add a field to the Task Revision family to store the revision number.

   > **Note:** The baseline Task Revision families already contain the Revision Number field that serves this purpose.

5. Run a script to modify the physical table of the Task Revision family.

6. Compile the database.

As you can see from these steps, the steps that you must complete are slightly different depending on whether you want to track changes to Task, Calibration Task, and Inspection Task records or records in some other Task family. The following lists separate the steps required in each scenario.

Configuration steps required to track changes to Task, Calibration Task, or Inspection Task records

1. Configure the Has Task Revision relationship to include the Task, Calibration Task, or Inspection Task family as the predecessor and the Task Revision, Calibration Task Revision, or Inspection Task Revision family as the successor, respectively.

2. Compile the database.

Configuration steps required to track changes to records in a different Task family

1. Add a Task Revision subfamily to the desired Task family.

2. Configure the Has Task Revision relationship to include the desired Task family as the predecessor and its Task Revision subfamily as the successor.

3. Add a field to the Task Revision family to store the revision number.

4. Run a script to modify the physical table of the subfamily.

5. Compile the database.

Configuring the Has Task Revision Relationship

The Has Task Revision relationship family is used to link Task Revision records to Task records. If a relationship definition exists between a Task family and its Task Revision subfamily, when you create a Task record in that family, the Meridium APM system will automatically create a Task Revision record and link it to the Task record. The Task Revision record serves as an historical record of the Task data on a given date and time. If you later update the Task record, a new Task Revision record will be created.

Meridium APM assumes that you do not want a Task Revision record to be created when you create a new Task record or update an existing Task record. If, however, you do want these Task Revision records to be created, you will need to configure the Has Task Revision relationship to include the desired families. No relationship definitions are configured for this family in the baseline Meridium APM database.

The following table provides examples of relationship definitions that you might configure for the Has Task Revision relationship if you are using the root Task family or a customer-defined Task family.

Predecessor	Predecessor Cardinality	Successor	Successor Cardinality
Task	Zero Or One	Task Revision	Zero or Many
Customer-defined Task family	Zero Or One	Customer-defined Task Revision family (which must be a subfamily of the customer-defined Task family)	Zero or Many

Calibration Task Revision

The Meridium APM system assumes that you do not want a Calibration Task Revision record to be created when you create a new Calibration Task record or update an existing Calibration Task record. If, however, you do want these Calibration Task Revision records to be created, you will need to configure the Has Task Revision relationship to include the desired families. No relationship definitions are configured for this family in the baseline Calibration Management database module.

The following table provides examples of relationship definitions that you might configure for the Has Task Revision relationship if you are using the Calibration Task family or a customer-defined Calibration Task family.

Predecessor	Predecessor Cardinality	Successor	Successor Cardinality
Calibration Task	Zero Or One	Calibration Task Revision	Zero or Many
Customer-defined Calibration Task family	Zero Or One	Customer-defined Calibration Task Revision family (which must be a subfamily of the customer-defined Calibration Task family)	Zero or Many

Inspection Task Revision

The Meridium APM system assumes that you do not want an Inspection Task Revision record to be created when you create a new Inspection Task record or update an existing Inspection Task record. If, however, you do want these Inspection Task Revision records to be created, you will need to configure the Has Task Revision relationship to include the desired families. No relationship definitions are configured for this family in the baseline Inspection Management product.

The following table provides examples of relationship definitions that you might configure for the Has Task Revision relationship if you are using the Inspection Task family or a customer-defined Inspection Task family.

Predecessor	Predecessor Cardinality	Successor	Successor Cardinality
Inspection Task	Zero Or One	Inspection Task Revision	Zero or Many
Customer-defined Inspection Task family	Zero Or One	Customer-defined Inspection Task Revision family (which must be a subfamily of the customer-defined Inspection Task family)	Zero or Many

Creating a Field to Store the Revision Number

If you want the Meridium APM system to create Task Revision records that are linked to a record *other than* a Task, Calibration Task, or Inspection Task record, you will need to add a revision number field to the Task Revision family.

This field must meet the following conditions:

- It must be a numeric field.
- Its ID must be **[Task Revision Family ID]_REV_NUM_N**. For example, the ID of the Revision Number field in the Calibration Task Revision family is MI_CLBTSKRV_ REV_NUM_N, where MI_CLBTSKRV is the ID of the Calibration Task Revision family.

About Filtering the Task Types List in Task Records

In Task Management, the Task Types family stores Task Types records, where each Task Types record represents a task type that can be assigned to a Task record. When you create or modify a Task record, the Task Type field will contain a list of Task Types records that exist in the database. If desired, you can create rules to filter this list so that it displays only the Task Types records that have been defined for the current Task family.

Meridium APM is preconfigured to filter the Task Type list in Calibration Task and Inspection Task records. For example, if you are creating an Inspection Task record, the Task Type list will display only the Task Types records that have been defined for Inspection Tasks.

Although Meridium APM is not preconfigured to filter the Task Type list in any other Task record, you can complete the following configurations steps to enable this functionality in other Task records:

- Create the desired Task family, or identify the desired baseline Task family.
- Add a System Code to the MI_TASK_TYPE_REFERENCE table.
- Define a Valid Values rule on the Task Type field of the appropriate Task family.
- Create the Task Types records for the Task families. After you do so, in Task records, the Task Type field will contain a list of Task Type records that exist in the database. This list may be filtered if a rule has been defined on the Task Type field for a given Task family. The baseline Meridium APM database contains some Task Types records in the Calibration Task, Inspection Task, and RCM FMEA Task families. You can create additional Task Types records in these and other Task families, if needed.

About the MI_TASK_TYPE_REFERENCE Table

The MI_TASK_TYPE_REFERENCE System Code Table contains System Codes that will populate the Reference field in each Task Types record. By default, three System Codes are defined in the MI_TASK_TYPE_REFERENCE System Code table:

- CALIBRATION
- INSPECTION
- RCM

Therefore, when you create that Task Types record, the Reference field will contain a list with three default values: CALIBRATION, INSPECTION, and RCM. By adding System Codes to the MI_TASK_TYPE_REFERENCE System Code table, you will add values to the list that appears in the Reference field of Task Types records.

The Reference field of a Task Types record allows you to associate that task type with a particular Task family. For example, Task Types records where the Reference field is set to CALIBRATION are associated with the Calibration Task family.

After you add System Codes to the MI_TASK_TYPE_REFERENCE System Code Table, you can define rules on each Task family so that the Task Type list will be filtered to display only those Task Types records that are associated with that Task family via the Reference value.

> **Note:** Although RCM is included in the MI_TASK_TYPE_REFERENCE System Code Table by default, there is NO default rule defined on the RCM FMEA Task family to filter the Task Types list using the RCM Reference value. To enable Task Type filtering on the RCM FMEA Task family, you will need to define a rule on the RCM FMEA Task family.

Defining a Valid Values Rule on the Task Type Field

After you have set up your Task families and the associated Task Type System Codes, you can define a Valid Values rule on the Task Type field of each family to indicate that the Task Type list will return only the Task Types records that have a given Reference value. By default, rules are defined for the Calibration Task and Inspection Task families. The following is an example of an Inspection Task rule:

Public Overrides Function GetPickList() As DynamicPickList

Dim pickList As DynamicPickList

pickList = MyBase.CreatePickList(True)

Return TaskUtility.Task_FamilyBehaviors.Picklist_State.MI_TASK_TASK_TYPE_CHR(_

Me.CurrentEntity, _

pickList, _

"**INSPECTION**")

End Function

> **Note:** In the rule, the specified reference must match exactly the ID of the System Code that you want to use for filtering the Task Type list. For example, specifying the reference value **INSPECTION** will cause the Task Type list to display only the Task Types records where the Reference value is **INSPECTION**.

Default rules are defined on the Inspection Task and Calibration Task families. If desired, you can define similar rules on additional Task families. Alternatively, you could define a rule at the Task family level and spread the rule to ALL Task subfamilies.

About Identifying Equipment and Functional Locations

Task records can be linked to multiple predecessor records that belong to different families. When you select a Task record that is linked to multiple predecessor records and then click any of the following links on the **Asset Options** menu, the Meridium APM system must find the predecessor record that identifies an *equipment or functional location* so that the action can be completed:

- **Show Recommendations:** In this case, the **Recommendation Management** page appears, which displays a list of Recommendation records that are linked to the Equipment or Functional Location record to which the selected Task record is linked. In this case, the Meridium APM system needs to identify the Equipment or Functional Location record to which the Task record is linked to determine which Recommendation records to show you.

- **Show Executions:** In this case, the **Task Execution History** page appears, which displays executions of ALL Task records that are linked to the Equipment or Functional Location record to which the selected Task record is linked. In this case, the Meridium APM system needs to identify the Equipment or Functional Location record to which the selected Task record is linked to determine which executions to show you.

- **Show Revisions:** In this case, the **Task Revision History** page appears, which displays revisions of ALL Task records that are linked to the Equipment or Functional Location record to which the selected Task record is linked. In this case, the Meridium APM system needs to identify the Equipment or Functional Location record to which the selected Task record is linked to determine which revisions to show you.

About the Equipment Asset Query and Location Asset Query

To identify the Equipment or Functional Location record that will be used to complete the operation, the Meridium APM system runs the following queries, which are stored in the Catalog folder \\Public\Meridium\Modules\Core\Queries:

- **Equipment Asset Query:** After providing a prompt for an Entity Key, the Equipment Asset Query returns the record with the provided Entity Key. In the baseline Meridium APM database, the query returns records in the Equipment family.

- **Location Asset Query:** After providing a prompt for an Entity Key, the Location Asset Query returns the record with the provided Entity Key. In the baseline Meridium APM database, the query returns records in the Functional Location family.

The Equipment Asset Query is always run first. If it returns a record, that record is assumed to be the correct predecessor record, and the Location Asset Query is not run. This means that if a Task record is linked to an Equipment record and a Functional Location record, the Equipment record will always be assumed to be the correct predecessor record.

You will need to modify the baseline queries ONLY if your Task records are linked to records in a family other than the Equipment family or the Functional Location family. When you modify the queries, keep in mind that they must contain the Entity Key field from the source family whose records you want to return. A prompt must exist on the Entity Key field, but the prompt can be a simple prompt with no valid values.

Both of these queries should query at the *highest* level necessary to include all equipment or location subfamilies. This means that if ALL of your customer-defined equipment and functional location families are structured in a hierarchy under a single parent family, such as *Asset,* you should modify *only* the Equipment Asset Query to include this parent family. In this case, the Location Asset Query will never be run because all customer-defined equipment and functional location records will be returned by the Equipment Asset Query.

When These Queries are Run

The Equipment Asset Query and Location Asset Query are not meant to be run on their own. Instead, the Meridium APM system runs these queries automatically when you select a Task record and then click any of the following links on the **Asset Options** menu:

- **Show Recommendations**
- **Show Executions**
- **Show Revisions**

When these queries are run, the Meridium APM system supplies the prompt with the Entity Keys of the predecessor records that are linked to the selected Task record using the following workflow:

- The Meridium APM system first runs the Equipment Asset Query. If any of the predecessor Entity Keys identify a record that belongs to the Equipment Asset Query's source family, the next page will be displayed in context of the record that is returned by the query.

- If none of the predecessor Entity Keys identify a record that belongs to the Equipment Asset Query's source family, the Meridium APM system runs the Location Asset Query. If any of the Entity Keys identify a record that belongs to this query source's family, the next page will be displayed in context of the record that is returned by this query.

- If none of the predecessor Entity Keys identify a record that belongs to the Location Asset Query's source family, the next page will not appear.

Overview of the Task Management Administration Page

You can use the **Task Management Administration** page to:

- [Define the Recommendation Query](#).
- [Define the Task Query](#).

> **Note:** You can access the **Task Management Administration** page only if you are a member of the MI Task Management Administrator Security Group.

Accessing the Task Management Administration Page

To access the Task Management Administration page:

- In the Meridium APM Framework, on the **Task Management Start Page**, click the **Administrative Tasks** link.

  The **Task Management Administration page** appears.
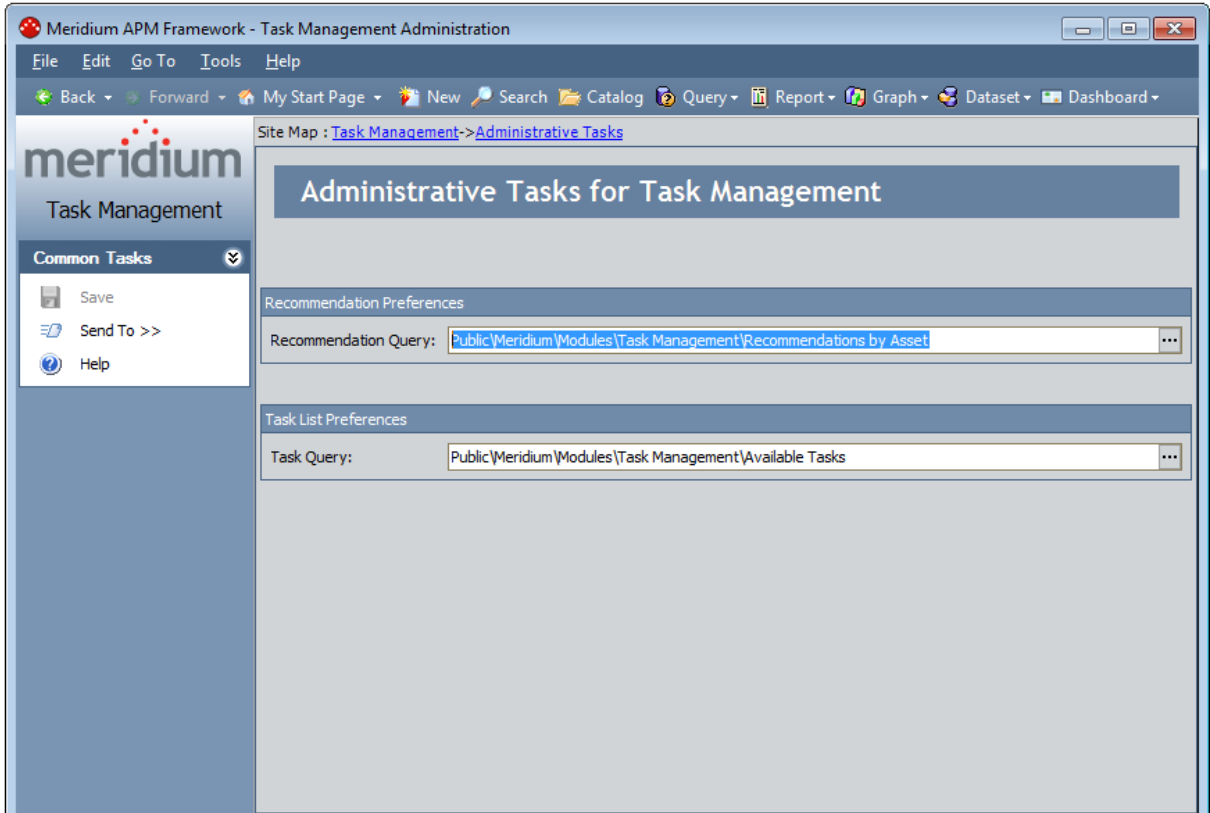
Aspects of the Task Management Administration Page

The **Task Management Administration** page contains the **Administrative Tasks for Task Management** workspace, which you can use to define a Recommendation query and a Task query. The **Administrative Tasks for Task Management** workspace is shown in the following image.
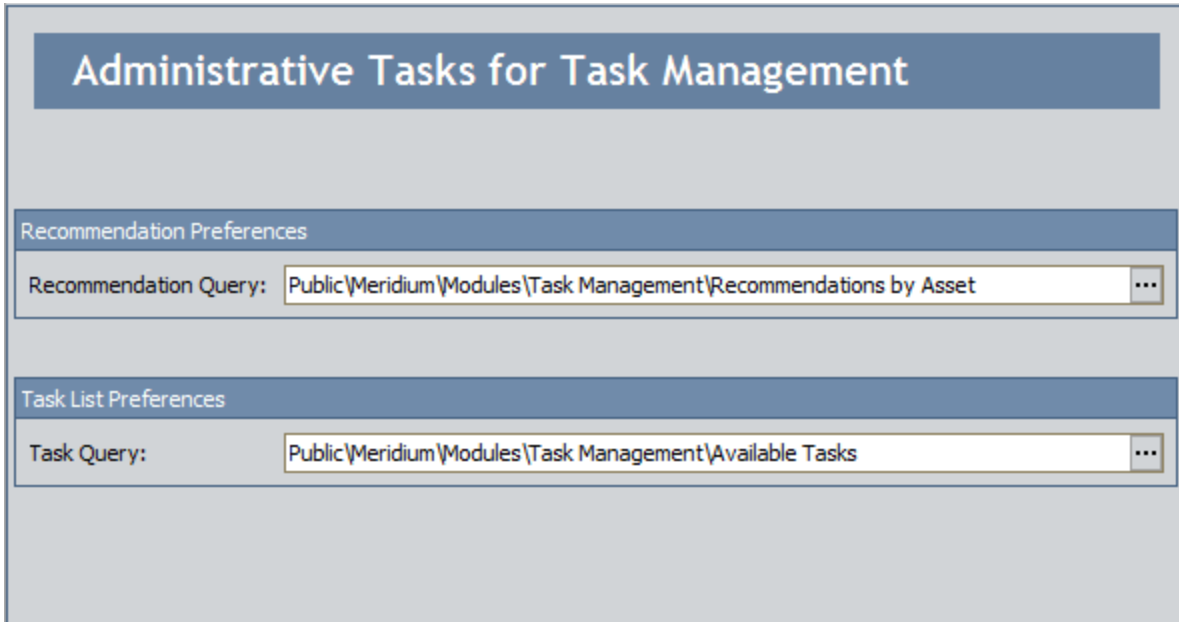
The **Task Management Administration** page contains one task menu, Common Tasks.
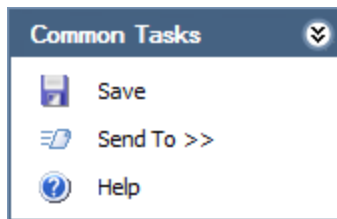
Common Tasks Menu

The **Common Tasks** menu on the **Task Management Administration** page contains the following links:

- **Save:** Saves any changes that you made to the selection in the **Recommendation Query** or **Task Query** cells. This link is enabled only after you have made a change.

- **Send To:** Displays a submenu with options that let you provide a link to the current page on your desktop (create shortcut), in an email message, or on a Home Page.

- **Help:** Displays the context-sensitive Help topic for the **Task Management Administration** page.

About the Recommendation Query

When you are viewing the **Task List** page, the **Asset Options** menu displays the **Show Recommendations** link. When you click the link, the Recommendation query runs and retrieve a list of Recommendation records that are linked to the Equipment or Functional Location record that is linked to the Task record that was selected on the **Task List** page.

For example, on the **Task List** page, if you select a row containing a Task record that is linked to an Equipment record, you can click the **Show Recommendations** link to view all Recommendation records that are linked to that Equipment record.

Meridium APM provides a default Recommendation query called Recommendations by Asset, which is stored in the following folder in the Catalog: \\Public\Meridium\Modules\Task Management. This query will be used automatically by Task Management. If you prefer to create your own query and use it instead, you will need to select that query on the **Task Management Administration** page.

> **Note:** If desired, you can customize the default query without having to modify the administrative settings. As long as you do not change the query name or location, the default query will be used automatically even if you change the content.

Defining the Recommendation Query

The baseline Meridium APM database contains a default Recommendation query called Recommendations by Asset, which is stored in the following folder in the Catalog: \\Public\Meridium\Modules\Task Management. This query will be used automatically by Task Management. If you prefer to create your own query and use it instead, you can use the following instructions to select that query on the **Task Management Administrative Tasks** page.

To define the Recommendation query:

1. In the Meridium APM Framework, on the **Task Management Administration** page, in the **Recommendation Query** text box, click the ⋯ button.

   The **Open Query** dialog box appears.

2. In the Catalog, navigate to the query that you want to use as the Recommendation query.

3. Select the query, and click the **Open** button.

   The path and query name appear in the **Recommendation Query** cell on the **Task ManagementAdministration** page.

4. On the **Common Tasks** menu, click the **Save** link to save your selection.

   The selection is saved, and the specified query will now be run when you access the **Recommendation Management** page from the **Task List** page.

About the Task Query

When you access the **Task List** page, the Meridium APM system runs the Task query to retrieve a list of all Task records that exist in the database.

The baseline Meridium APM database contains a default Task query called Available Tasks, which is stored in the following folder in the Catalog: \\Public\Meridium\Modules\Task Management. This query will be used automatically by Task Management.

The Available Tasks query contains two prompts: Asset ID and Task Family. When you are viewing the **Task List** page, the prompts are displayed above the results so that you can modify the prompt values and rerun the query as desired.

If you prefer to create your own query and use it instead, you will need to select that query on the **Task Management Administration** page.

> **Note:** If desired, you can customize the default query without having to modify the administrative settings. As long as you do not change the query name or location, the default query will be used automatically even if you change the content.

Defining the Task Query

The baseline Meridium APM database contains a default Task query called Available Tasks, which is stored in the following folder in the Catalog: \\Public\Meridium\Modules\Task Management. This query will be used automatically by Task Management. If you prefer to create your own query and use it instead, you can use the following instructions to select that query on the **Task Management Administration** page.

To define the Recommendation query:

1. In the Meridium APM Framework, on the [Task Management Administration page](#), in the **TaskQuery** cell, click the ⋯ button.

   The **Open Query** dialog box appears.

2. In the Catalog, navigate to the query that you want to use as the Task query.

3. Select the query, and click the **Open** button.

   The path and query name appear in the **Task Query** cell on the **Task ManagementAdministration** page.

4. On the **Common Tasks** menu, click the **Save** link to save your selection.
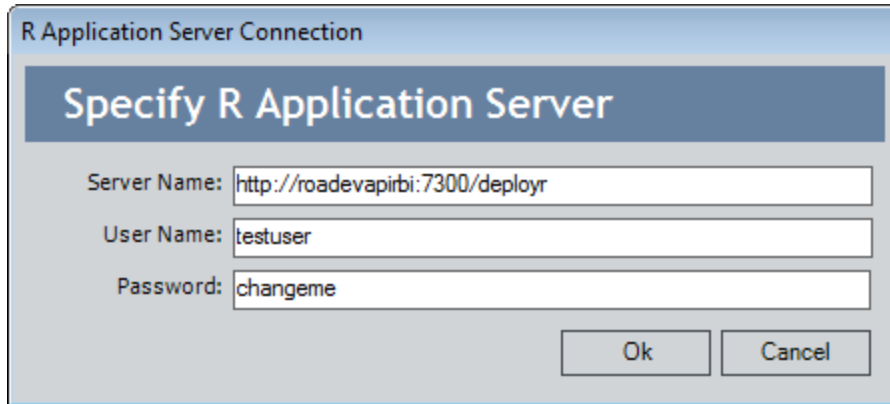
   The selection is saved, and the specified query will now be run when you access **Task List** page.

R Script System Requirements

This feature is available with the core Meridium APM Framework application; no specific license is required.

In addition to the basic Meridium APM Framework application, you must also deploy R Servers. In addition to deploying an R Server on your Meridium APM Application Server (or a dedicated R Server), you must also install an R Server on *each* Meridium APM Policy Execution Server. You must also specify R Server credentials in the Configuration Manager. The following diagram illustrates the architecture.

R Server

A machine on which the DeployR software, and its dependencies, is installed. The software requirements of this server are determined by Revolution Analytics, the third-party distributor of the software. You can find more information about this software, including required software dependencies and installation instructions, on the Revolution Analytics website.

> **IMPORTANT:** The World Wide Web Publishing service must be *enabled* on the R Server in order to use R scripts in the Meridium APM Framework application. If this setting is disabled during the installation of DeployR, you must enable it.

Configuring R Server Credentials

To configure your R Server credentials:

1. Open the Configuration Manager application.

2. On the Configuration Manager main menu, click **Tools**, point to **Global Preferences**, and then click **R-Engine Server Configuration**.

   The **R Application Server Connection** dialog box appears.

3. In the **Server Name** text box, type the name or IP address of the server on which DeployR is installed.

   IMPORTANT: You must structure the address in the Server Name box in the following way: http://<server name or IP address>:<port number>/deployr. <Server name or IP address> corresponds to the Meridium APM Application Server or dedicated R Server where DeployR is installed. <Port number> is defined by the version of DeployR you are using. Reference the DeployR documentation for more information.

4. In the **User Name** text box, type a user name that corresponds to a username that exists for your DeployR server.

5. In the **Password** text box, type the password that corresponds to the username.

   The **R Application Server Connection** dialog box closes. R Scripts will now function in the Meridium APM Framework application.

Accessing the Dashboard Page

While access to baseline dashboards is provided in various ways depending upon the module or work process that utilizes the dashboard, you can access all dashboards via the Meridium APM Catalog, as described in the following instructions.

To access the Dashboard page:

1. In the Meridium APM Catalog, navigate to the location of the dashboard that you want to view on the **Dashboard** page.

2. Double-click the dashboard that you want to open.

   -or-

   Right-click the dashboard, and click **Open**.

   The dashboard is displayed on the **Dashboard** page. The following image shows an example of the **Dashboard** page.
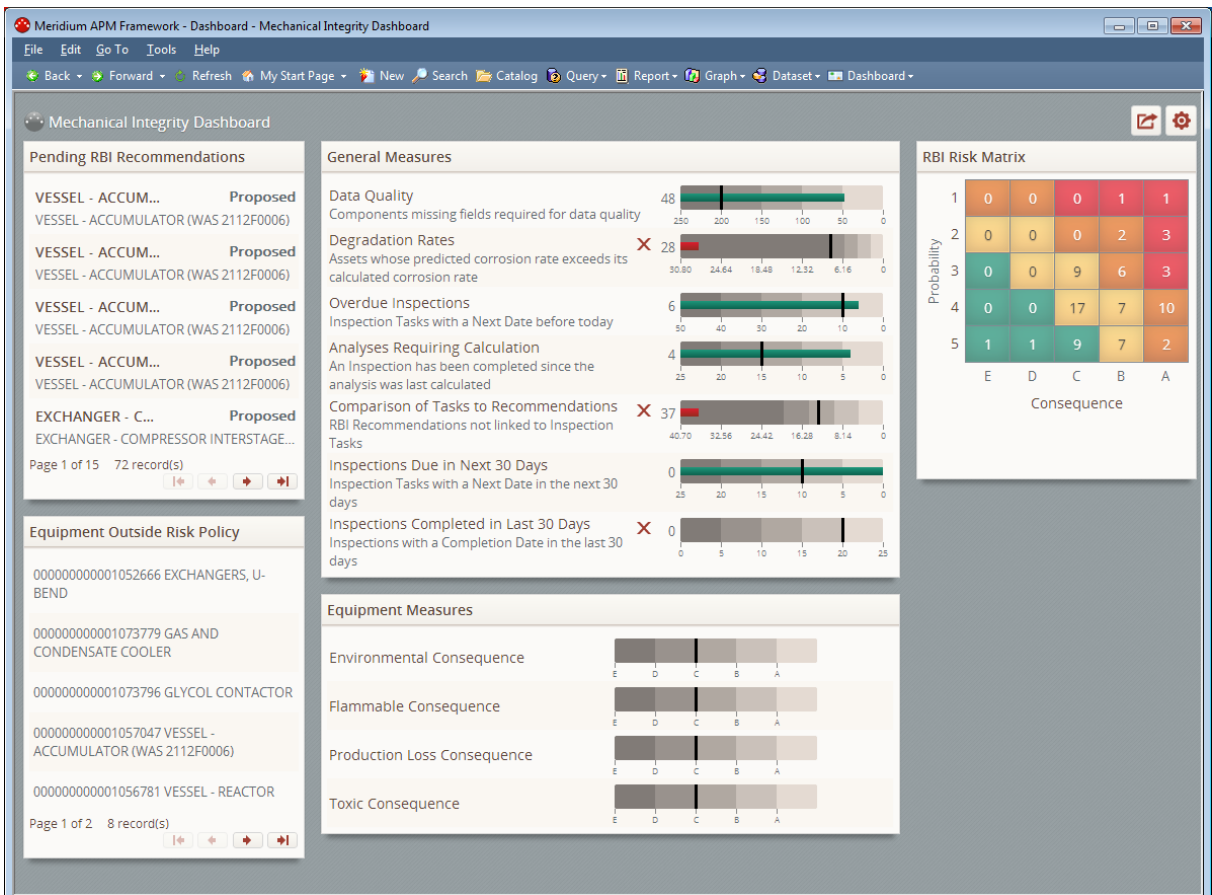
Aspects of the Dashboard Page

The **Dashboard <Catalog item caption>** page, where **<Catalog item caption>** is the dashboard Catalog item caption, displays various sections that have been configured for a given dashboard. Throughout this documentation, we refer to the **Dashboard <Catalog item caption>** page simply as the *Dashboard page*. The following image shows an example of the **Dashboard** page.
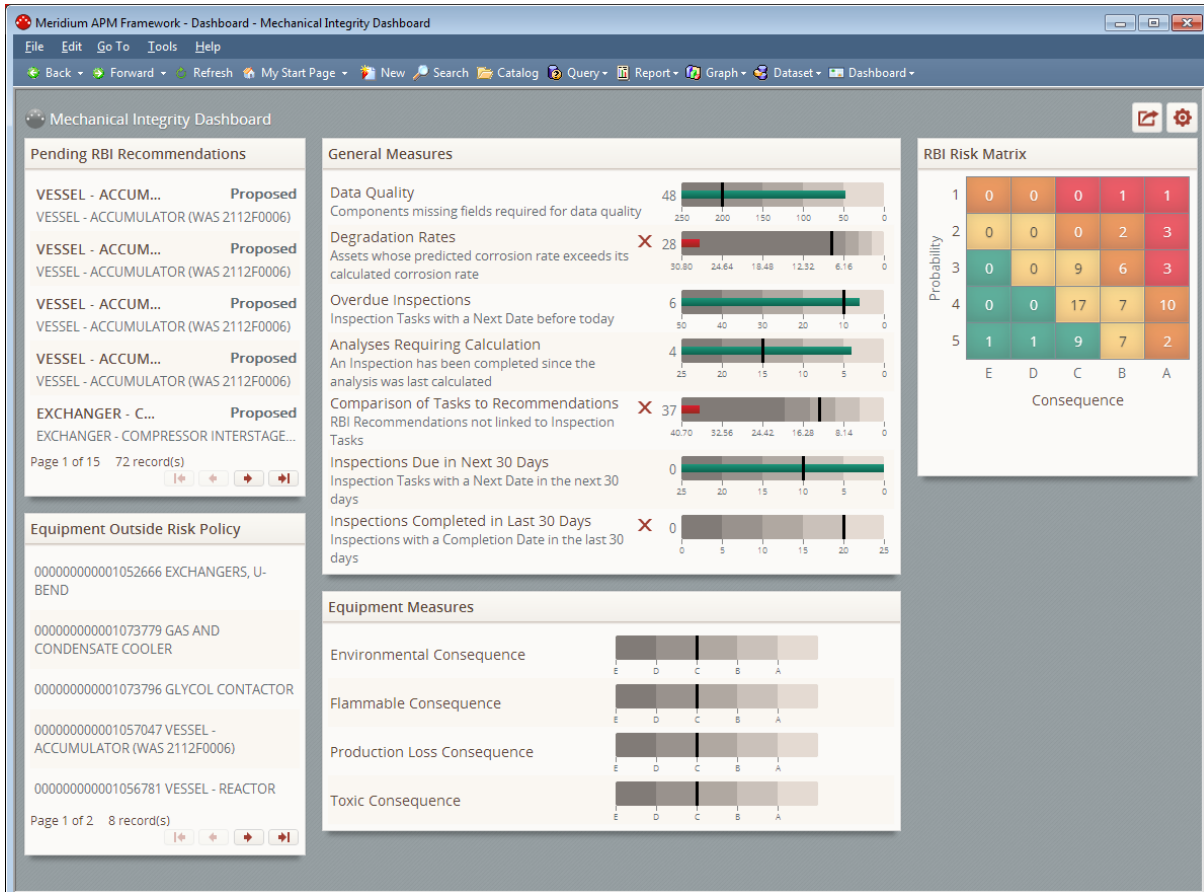
The **Dashboard** page displays the following items:

- **Text of the Catalog location of the dashboard:** Indicates the location of the dashboard in the Catalog. This text is displayed on the left side of the page, below the Meridium APM Framework toolbar. In the above image, the text **Public\Meridium\Modules\Mechanical Integrity\Dashboards\MI Dashboard** is displayed.

- button:Creates an email message that contains a link to the dashboard.

- button:Displays the dashboard configuration buttons.

- **Dashboard sections:** Displays one or more sections depending upon how the dashboard has been configured. In the above image, the following sections are displayed on the dashboard:

- Pending Recommendations
- Equipment Outside Risk Policy
- General Measures
- Equipment Measures
- RBI Risk Matrix

About Designing Dashboards

Dashboards contain sections that can be used to display information about the data in your Meridium APM system and provide access to various areas in the Meridium APM Framework. If you have the [necessary security permissions](#), you can design a dashboard that displays various sections.

This section of the documentation provides information on:

- [Dashboard configuration buttons](#).
- [Dashboard sections](#).
- [Creating new dashboards](#).
- [Modifying existing dashboards](#).

About Dashboard Configuration Buttons

Depending upon your security permissions, you can access various configuration buttons on the **Dashboard** page. These buttons can be used to:

- Add sections to a dashboard.
- Remove sections from a dashboard.
- Specify the number of columns and/or column on a dashboard.
- Arrange the content of columns on a dashboard.
- Hide sections on a dashboard.
- Redisplay hidden sections on a dashboard.
- Modify the height of a dashboard section.
- Restore a dashboard section to its default size.
- Clear your user display preferences.

Callouts have been added to the following image to highlight the configuration buttons that are displayed on the **Dashboard** page.
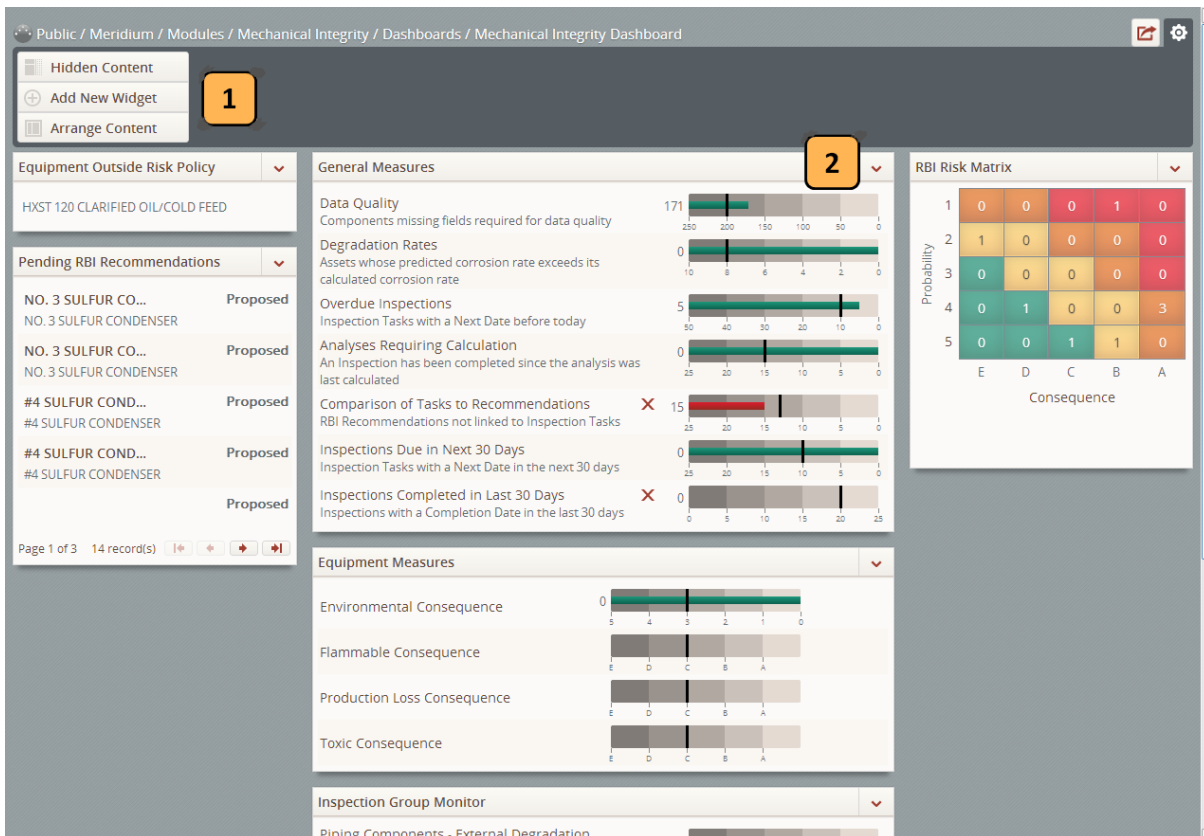
The following table lists the callouts that are displayed in the image and the corresponding configuration button(s).

Callout	Dashboard configuration button(s)
1	The **Hidden Content** and **Arrange Content** buttons are displayed above the dashboard sections, and depending upon your Catalog security permissions, one of the following additional buttons will be displayed: • **Add New Widget:** Displayed only to users who have Edit permissions to the Catalog folder where the dashboard is stored. • **Revert Changes:** Displayed to all users.
2	The ⌄ button is displayed at the top of each dashboard section. This button displays various options, depending upon your Catalog security permissions.

Additionally, when you click a configuration button, the ⚙ button changes to ⚙ .
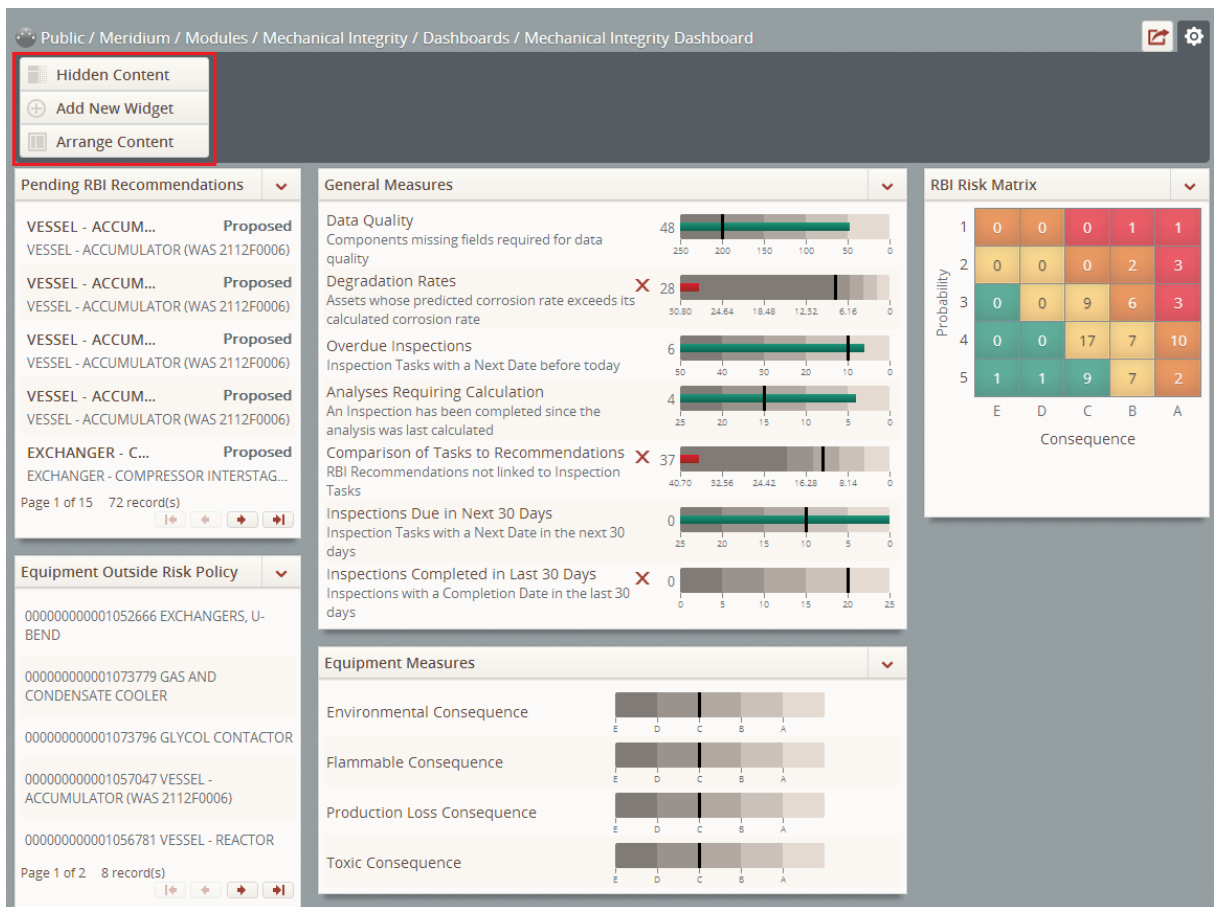
Accessing Dashboard Configuration Buttons

The following instructions assume that you are viewing the dashboard for which you want to access the dashboard configuration buttons.

To access the configuration buttons:

- Click the button.

  Depending upon your security permissions, various configuration buttons are displayed. A red outline has been added to the following image to highlight the buttons that are displayed to a user who has the security permissions necessary to add sections to a dashboard.

Hidden Content Button

When you click **Hidden Content** button:

- A list appears, displaying the name of each dashboard section that is currently [hidden](#).
- Buttons appear that allow you to [redisplay hidden sections](#) on a dashboard.
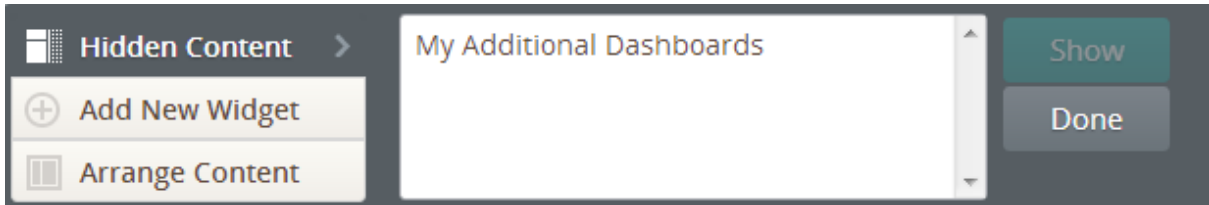
The following buttons are displayed to the right of thelist of hidden sections:

- **Show:** [Redisplays the section](#) on the dashboard that corresponds with the selected value in the **Hidden Content** list. This button is enabled only when a hidden section is selected in the list *and* when *fewer than* 12 sections are currently displayed on the dashboard.
- **Done:** Hides the **Hidden Content** list and the **Show** and **Done** buttons.

Add New Widget Button

When you click the **Add New Widget** button, the list of dashboard sections and buttons that are used to add new sections to dashboards are displayed.
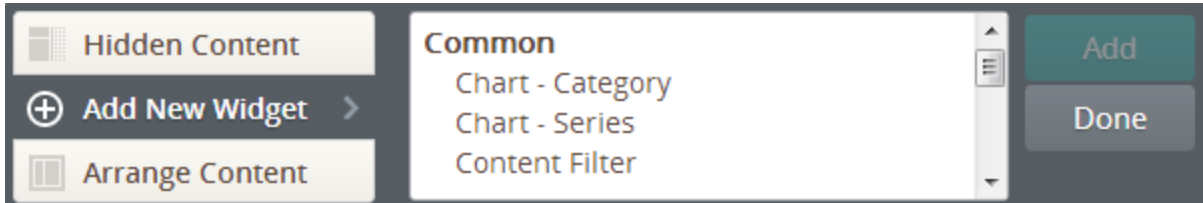
The **Add New Widget** button is displayed only to users who have Edit privileges to the Catalog folder that stores the dashboard. The list of dashboard sections contains the following headings:

- **Common:** Lists the following dashboard sections that are common to all dashboards:
    - Chart - Category
    - Chart - Series
    - Content Filter
    - Hyperlinks
    - List - Query Result
    - List - Single Column
    - List - Summary
    - Measures - General
    - Measures - Linked
- **Mechanical Integrity:** Lists the dashboard section **RBI Risk Matrix**, which is specific to the baseline Mechanical Integrity dashboard and dashboards that are used by Mechanical Integrity users when the Risk Based Inspection license is active.

The following buttons are displayed to the right of the dashboard section list:

- **Add:** Displays the selected section on the dashboard in edit view, which you can use to define the content that will be displayed in the section. This button is enabled only when a value is selected in the list of dashboard sections *and* when *fewer than* 12 sections are currently displayed on the dashboard.
- **Done:** Hides the list of sections and the **Add** and **Done** buttons.

Arrange Content Button

When you click the **Arrange Content** button, options for modifying the appearance of dashboard content are displayed, as shown in the following image.
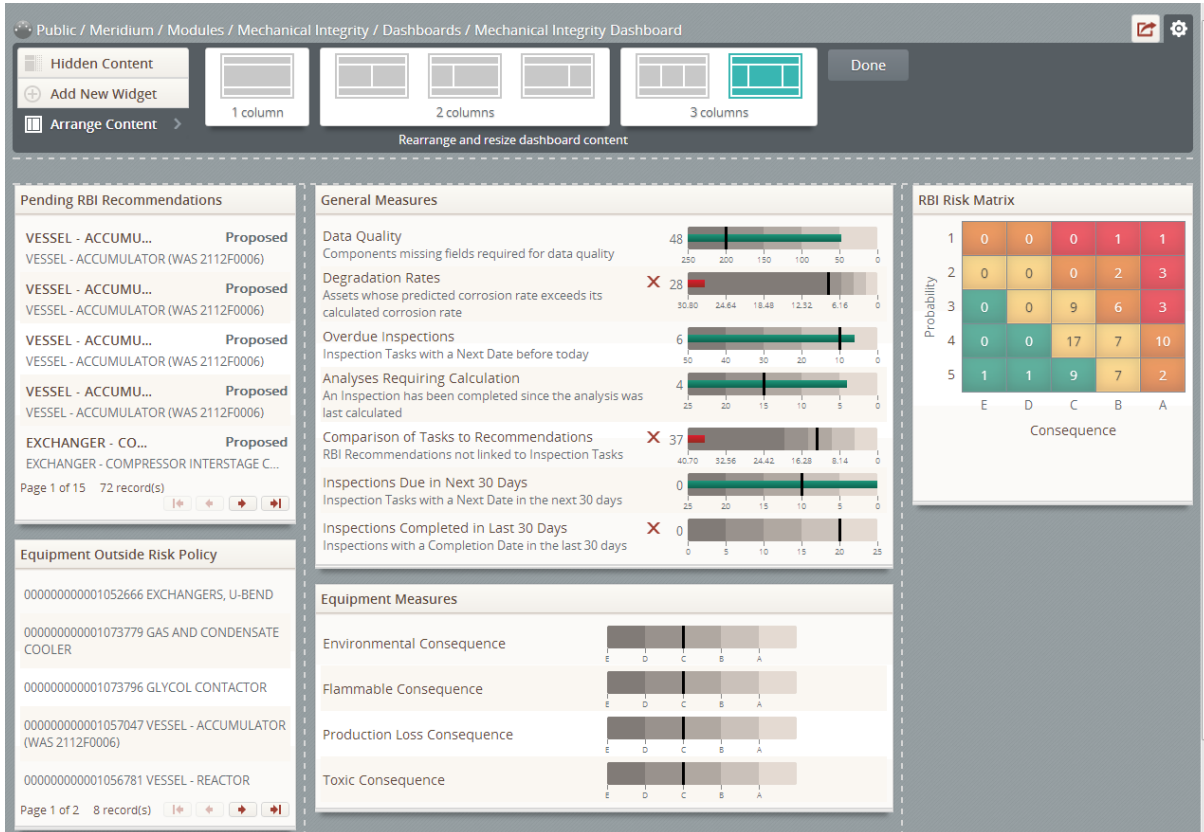
When you click the **Arrange Content** button, the following items appear:

- **Column options:** Displays the following options for the number of columns that are displayed in addition to the top and bottom rows on the dashboard:
    - **1 column:** One option for displaying the content of your dashboard in the top and bottom rows and in one column.
    - **2 columns:** Three options for displaying the content of your dashboard in the top and bottom rows and two columns.
    - **3 columns:** Two options for displaying the content of your dashboard in the top and bottom rows and three columns.

Regardless of the column option that you select, one row is always displayed at the top and bottom of the dashboard that spans the width of the page. If you do not specify that at least one section should be displayed in either of these rows, the rows will be hidden when you click the **Done** button. You can arrange dashboard sections in these rows just as you would arrange dashboard sections within the columns.

Note: The dashboard may display columns with widths that are *different* from what is shown by the column option that you select, depending upon the length of words displayed in sections in a given column. For example, if you select the three column option for three, equal width columns but one of your sections contains a word whose number of characters *exceeds* with width of the column, three, equal width columns will *not* be displayed. The width of the column containing the word whose number of characters exceeds with width of the column will be *larger* than the width of the other two columns. To avoid this issue, try selecting a different column layout option.

Dotted lines: Used to denote the current column and row layout of the dashboard that can be used to arrange dashboard sections within the columns and rows.

- **Done button:** Displayed to the right of the column options, which hides the column options and the **Done** button.

Revert Changes Button

The **Revert Changes** button, shown in the following image, is displayed only to users who do *not* have Edit privileges to the Catalog folder that stores the dashboard.
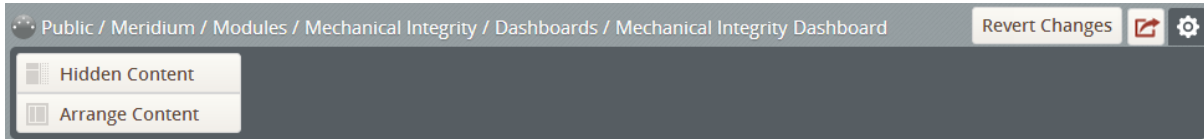
When you click the **Revert Changes** button:

- A confirmation message appears, and any changes that you have made to the appearance of sections on the dashboard that you are working with are reversed.
- The dashboard configuration buttons are hidden.

Dashboard Section Button

When you access the dashboard configuration buttons, the ⌄ button appears at the top of each section that is displayed on the dashboard. A red outline has been added to the following image to highlight the ⌄ button.
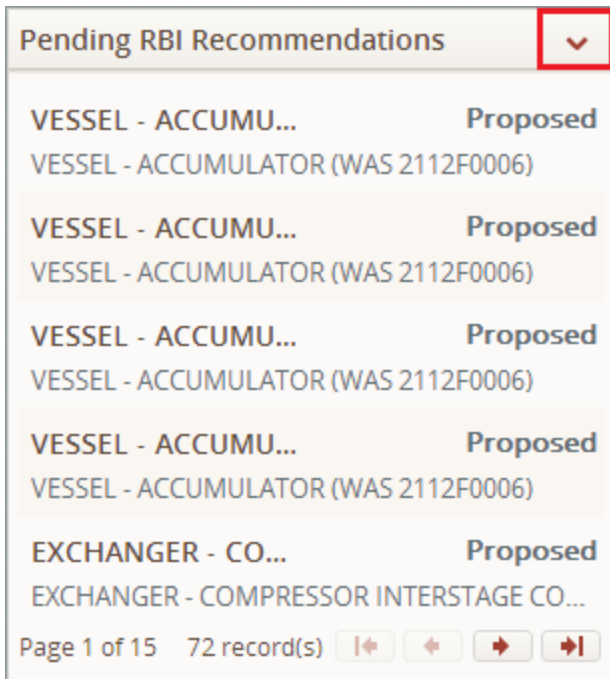
When you click the ⌄ button, the following submenu is displayed.
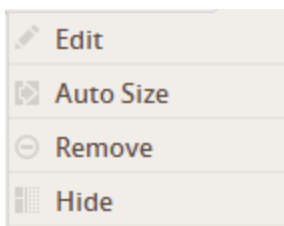
The submenu contains the following options:

- **Edit:** Displays the edit view of the section. This option is displayed only for users that have Edit privileges to the Catalog folder that stores the dashboard.
- **Auto Size:** Restores the dashboard section to the size that it is displayed by default.
- **Remove:** After displaying a confirmation message, removes the section from the dashboard. This option is displayed only for users that have Edit privileges to the Catalog folder that stores the dashboard.
- **Hide:** Hides the section on the dashboard.

About Creating Dashboards

If you have the necessary security permissions, you can create dashboards from scratch or from a copy of an existing dashboard. You would create a new dashboard:

- From scratch if a dashboard from which you want to create a copy does not exist.
- As a copy of an existing dashboard if a dashboard exists that is similar to the dashboard that you want to create. To do this, you would simply create a copy of the dashboard Catalog item. When you create a dashboard as a copy of an existing dashboard, the Catalog items are not related in any way. Any changes or updates to the source or copied dashboard will be reflected only on the dashboard for which the changes or updates were made.

Once you have created a dashboard, depending upon the information that you want it to display or how other users will use the dashboard, you can:

- Add sections to the dashboard.
- Remove sections from the dashboard.
- Modify the content of the dashboard sections.
- Perform various actions to modify the display of the dashboard.

Creating a Dashboard From Scratch

You can create a dashboard from scratch if a dashboard from which you want to create a copy does not exist.

To create a dashboard from scratch:

1. On any page in the Meridium APM Framework, on the Meridium APM Framework toolbar, click the **Dashboard** button, and then click the **Build New** option.

   The **Save Dashboard** dialog box appears.
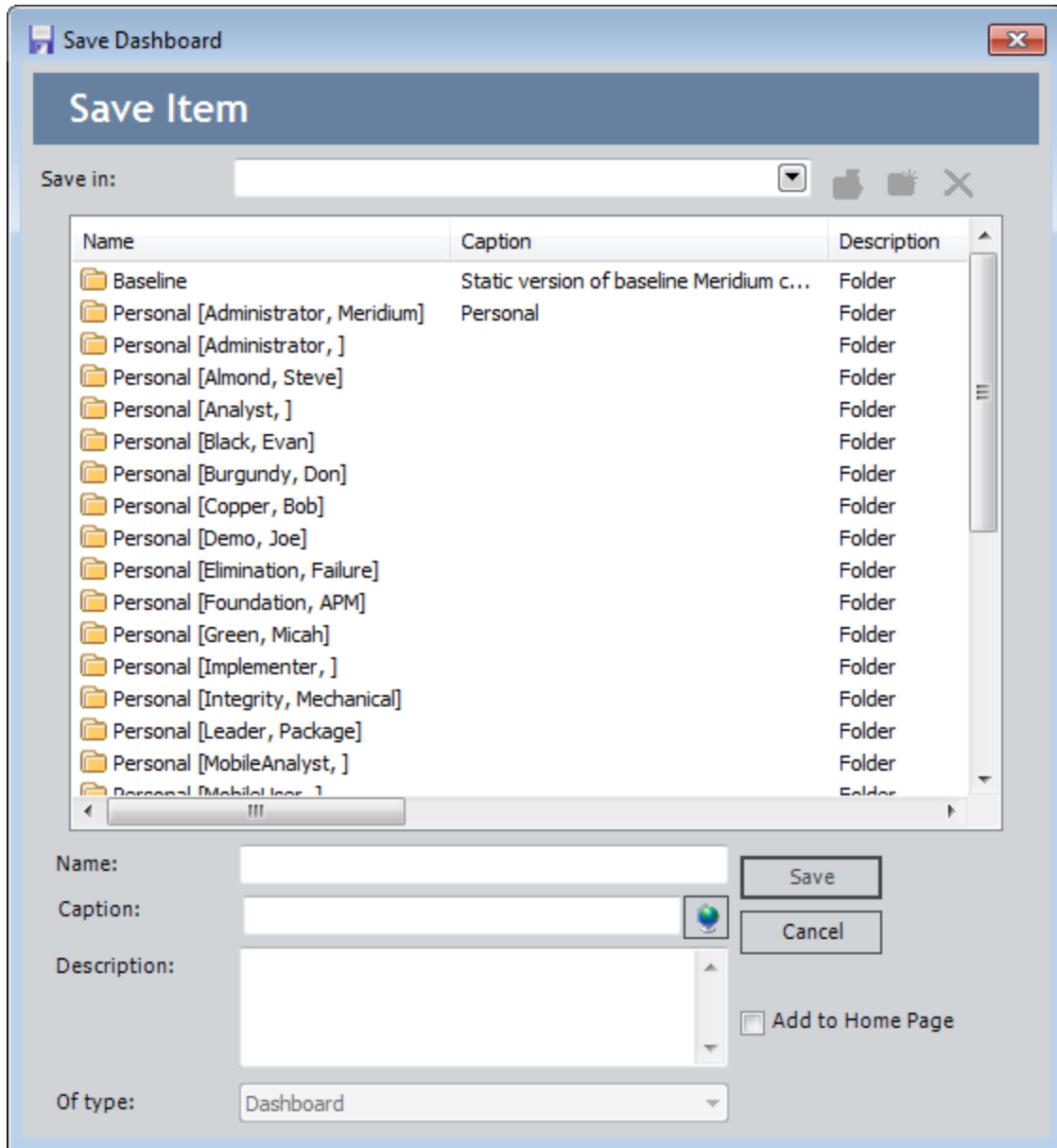
The **Of type** list is disabled and contains the value *Dashboard* by default. You cannot modify this selection.

2. Navigate to the Catalog location to which you want to save the new dashboard.

3. In the **Name** text box, type a name for the dashboard. The name is required and must be unique to the Catalog folder in which you are saving the dashboard.

    The **Caption** text box is populated automatically with the value that you typed in the **Name** text box.

4. If desired, click the ![globe icon] button to manage the translations for that string.

5. If desired, type a description of the dashboard in the **Description** text box. This is not required to save the dashboard.

6. If desired, select the **Add to Home Page** check box if you want to add a link to the dashboard to a Home Page. This is not required to save the dashboard.

7. Click the **Save** button.

    If you:

- Selected the **Add to Home Page** check box, the Send to Home Page - Select Home Page dialog box is displayed.
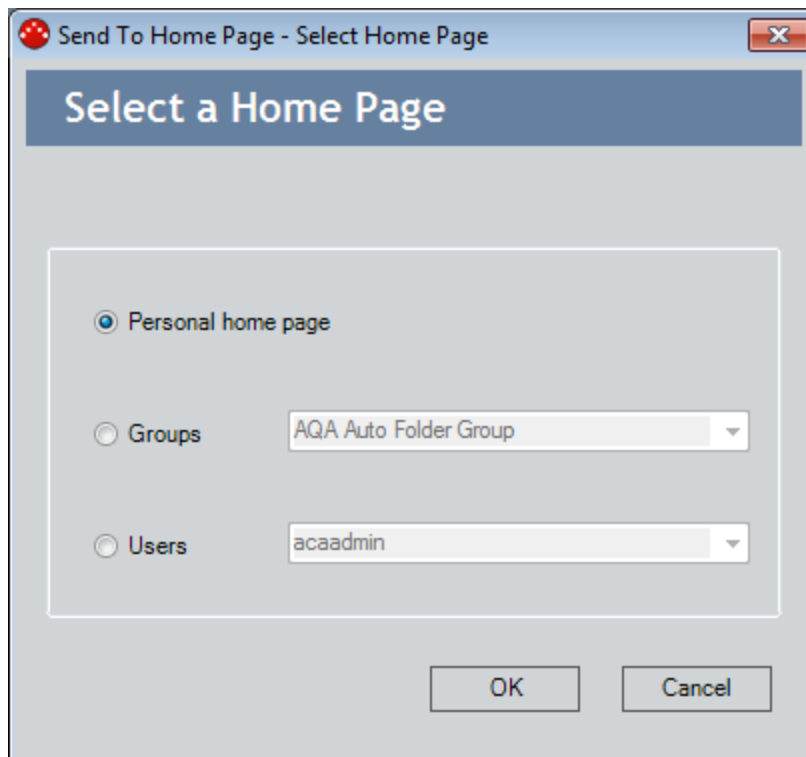
    Depending upon your membership in the MI Power User Role Security Group and whether or not you are a Super User, you can use the **Send to Home Page - Select Home Page** dialog box provide a link to the dashboard

to your Personal Home Page, a Group Home Page, or the Personal Home Page for a particular Security User.

- Did not select the **Add to Home Page** check box, or you have completed the process of adding the dashboard to a Home Page, the **Save Dashboard** dialog box closes, and a blank **Dashboard** page is displayed.
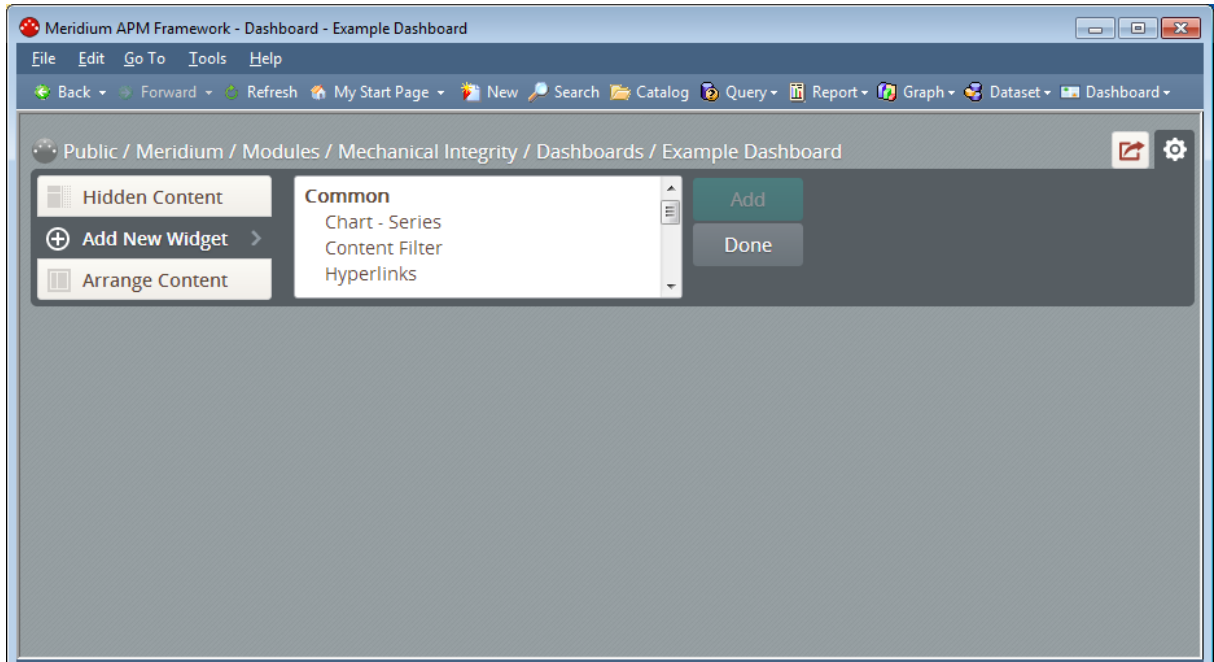
By default, the [dashboard configuration buttons are displayed](#), and the **Add New Widget** button is selected, displaying the list of dashboard sections and buttons that are used to [add sections to a dashboard](#).

About Dashboard Sections

The content of a dashboard is displayed in various sections on the [Dashboard page](#). The number of sections displayed on the page and their content depends upon information defined by the user who last saved the dashboard.

The following sections that are common to all dashboards are provided in the baseline database:

- [Chart - Category](#)
- [Chart - Series](#)
- [Content Filter](#)
- [Hyperlinks](#)
- [List - Query Result](#)
- [List - Single Column](#)
- [List - Summary](#)
- [Measures - General](#)
- [Measures - Linked](#)

One section, **RBI Risk Matrix**, is specific to the baseline Mechanical Integrity dashboard and dashboards that are used by Mechanical Integrity users when the Risk Based Inspection license is active. Because modifying this section is not recommended, documentation for working with the edit view of this section is not provided.

Each dashboard section has:

- **An edit view**, which can be accessed *only* by users that have Edit permissions to the Catalog folders that store a dashboard. In order for these users to make changes to the dashboard section via the edit view, they must be members in the appropriate module-level Security Groups.
- **A read-only view**, which is displayed to all users that are members of the appropriate module-level Security Groups.

Throughout this documentation, we use the term *dashboard section* or *section* to refer to the read-only view of a dashboard section. We use the term *edit view* or *edit view of the section* to refer to the edit view of a dashboard section.

About the Edit View of Dashboard Sections

The edit view of each dashboard section contains various items that are used to configure the content that is displayed in that dashboard section. The following items are common to the edit view of *all* dashboard sections:

- **Label text box:** An unlabeled text box that appears at the top of the edit view of each dashboard section. The text box contains the name of the dashboard section by default. The value that you type in this text box will be used to label the section on the dashboard.
- **Donebutton:** Displayed to the right of the label text box, this button closes the edit view of the dashboard section.

A red outline has been added to the following image of a **Hyperlinks** dashboard section to highlight the label text box and the **Done** button.
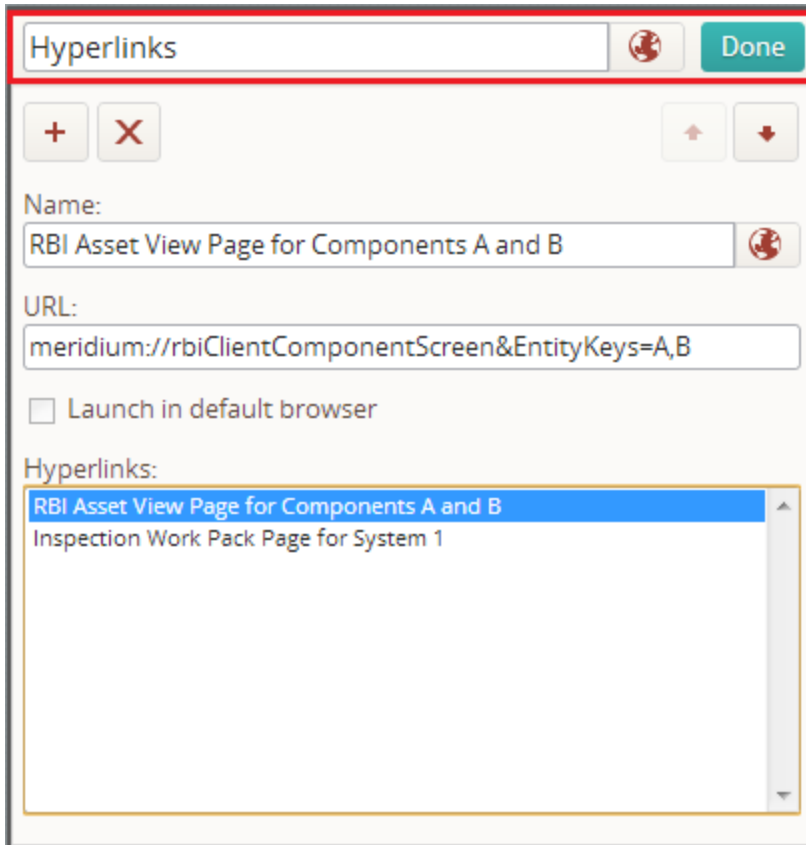
About Working with Dashboard Sections

If you have the necessary security permissions, you can:

- Add sections to a dashboard.
- Modify the content of existing dashboard sections.
- Remove sections from a dashboard.

Adding Sections to Dashboards

You can add sections to a dashboard using the configuration buttons on the Dashboard page. You can add as many sections as you want to a dashboard, however, a maximum of only 12 sections can be displayed on the dashboard at one time. For optimal performance and usability, however, we recommend that you display *no more than* eight sections on a dashboard at one time. These instructions assume that you have accessed the configuration buttons for the dashboard to which you want to add a section.

To add a section to a dashboard:

1. Click the **Add New Widget** button.

   The list of dashboard sections and the buttons used to add these sections to a dashboard are displayed.
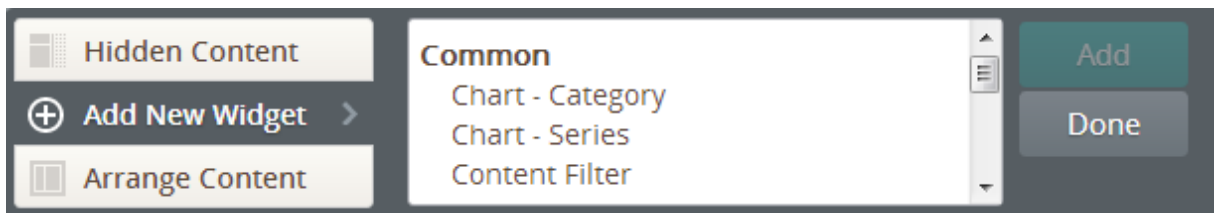
2. In the list of dashboard sections, select the name of the section that you want to add to the dashboard.

3. Click the **Add** button.

   The edit view of the new section is displayed at the top of the first column on the left side of the dashboard. The following image shows an example of the edit view of a **Hyperlinks** section that has been added to a dashboard.

> **Note:** If the dashboard to which you added a new section already displays between eight and 11 sections, a warning message appears to the right of the **Add** button, indicating that you should not add more than eight sections to a dashboard.

4. In the edit view of the dashboard section, specify the content that you want to appear in that section.

   The following image shows an example of the edit view of a **Hyperlinks** section for which two hyperlinks have been defined.

5. Click the **Done** button.

   The section is displayed on the dashboard.

Removing Sections From a Dashboard

The following instructions assume that you have accessed the configuration buttons on the dashboard from which you want to remove a section.

To remove a section from a dashboard:

1. At the top of the section that you want to remove, click the ⌄ button.

   A submenu appears. The following image shows an example of the submenu that is displayed after clicking the ⌄ button at the top of a **Hyperlinks** section on a dashboard.
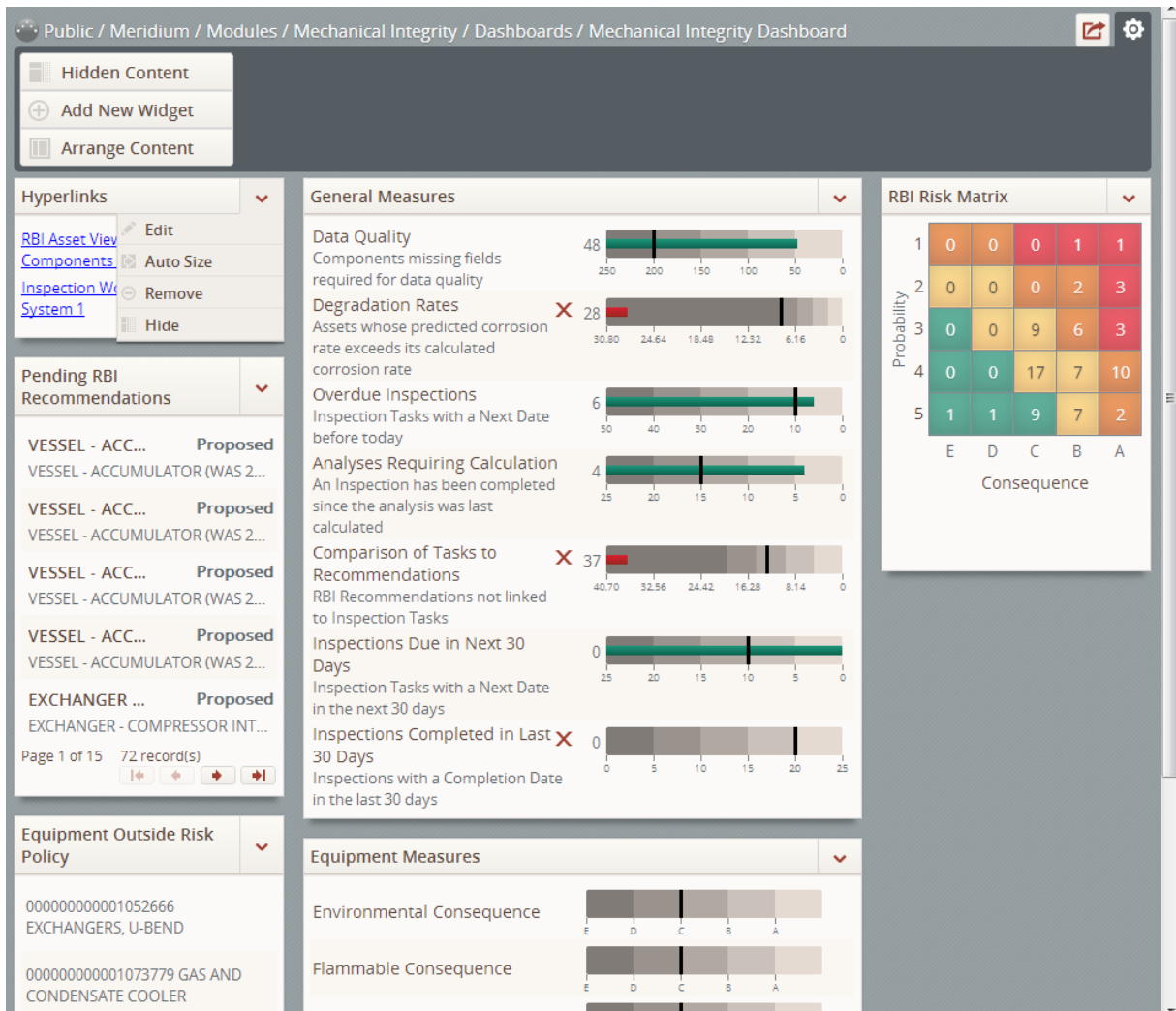
2. Click the **Remove** option.

   A confirmation message appears, asking if you are sure that you want to remove the section.

3. Click the **Remove** button.

The section is removed from the dashboard.

Modifying the Content of Existing Dashboard Sections

You can modify the content of an existing dashboard section using the edit view of that section. These instructions assume that you have accessed the configuration buttons on the dashboard whose section you want to modify.

To modify the content of an existing dashboard section:

1. At the top of the dashboard section that you want to modify, click the ⌄ button.

   A submenu appears. The following image shows an example of the submenu that is displayed after clicking the ⌄ button at the top of the **General Measures** section on a dashboard.

2. Click the **Edit** option.

   The edit view of the section is displayed. The following image shows an example of the edit view of the **General Measures** section.

3. Using the edit view, make the appropriate modifications. For example, a red outline has been added to the following image to highlight the following modifications:

    - The **Display Measure Description** check box in the **Display Options** sub-section has been *cleared*.
    - For the Data Quality measure, the value in the **Target** text box in the **Measure** sub-section has been changed from *200* to *250*.

4. Click the **Done** button.

   The edit view of the section closes, and the dashboard section displays your modifications.

General Measures

Measure	Value	Chart
Data Quality	48	
Degradation Rates	✗ 28	
Overdue Inspections	6	
Analyses Requiring Calculation	4	
Comparison of Tasks to Recommendations	✗ 37	
Inspections Due in Next 30 Days	0	
Inspections Completed in Last 30 Days	✗ 0	

About the Chart - Category Section

A **Chart - Category** section on a dashboard displays a graphical representation of query results, as specified in the [edit view of the **Chart - Category** section](). Specifically, a **Chart - Category** section displays a doughnut graph.

The following image shows an example of a **Chart - Category** section that has been configured for a dashboard.
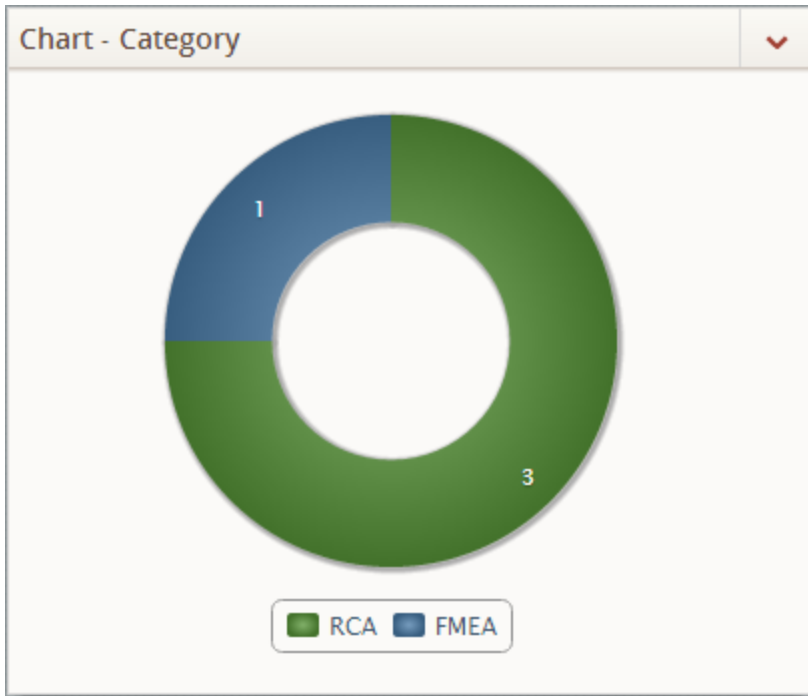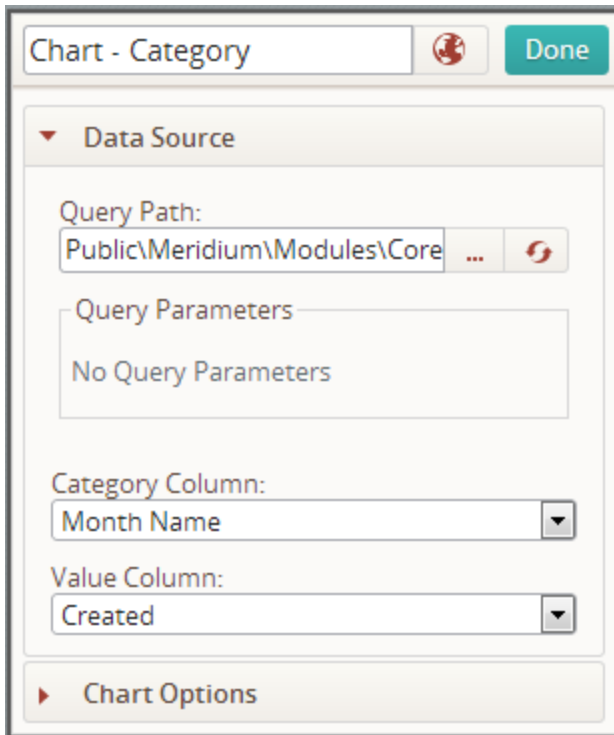
About the Edit View of the Chart - Category Section

In addition to the label text box and Done button, the edit view of the **Chart - Category** section displays the following sub-sections that are used to configure the doughnut graph that is displayed in a **Chart - Category** section on a dashboard:

- **Data Source** sub-section
- **Chart Options** sub-section

The following image shows an example of the edit view of a **Chart - Series** section that is configured for a dashboard.

Data Source Sub-Section

The **Data Source** sub-section is a collapsible sub-section that provides options for specifying the source query for values plotted on the doughnut graph in a [Chart - Category section](#) on the dashboard. When you access the edit view of the **Chart - Category** section, the **Data Source** sub-section is displayed by default.

The following image shows an example of the **Data Source** sub-section.

The **Data Source** sub-section contains the following items:

- **Query Pathtext box:** Contains the Catalog location of the query whose results will be plotted in the doughnut graph in the [Chart - Category section](#) (i.e., the source query). To the right of this text box, the following buttons are displayed:

  - ⬛ ... : Displays the **Choose Query** dialog box where you can select the source query from the Meridium APM Catalog. When you specify the source query in this way, if the selected query contains prompts, fields that correspond with these prompts appear in the **Query Parameters** pane.

  - ⬛ 🔄 : Refreshes the source query. If you specify the source query manually and the specified query contains prompts, you must click this button in order for the prompt fields to appear in the **Query Parameters** pane.

  If this text box does not contain a valid query path, a [validation message appears](#) and a red outline appears around this text box and the **Query Parameters** pane until you specify a valid query path.

- **Query Parameters pane:** If the source query contains prompts, this pane displays text boxes and lists where you should specify values for these prompts. The labels of the text boxes and lists correspond with the captions of prompt fields. If this pane contains lists and a value is not selected in each list, a [validation message appears](#) and a red outline appears around this text box and the **Query Path** text

box until you select a value in each list.

- **Category Column list:** Specifies the category that will be plotted in the doughnut graph. This list contains the captions of all fields returned by the source query. For example, the graph in the image that follows this list shows an example of a **Chart - Category** section whose category consists of the following values: *High, Medium, Low, Incomplete.*

- **Value Column list:** Specifies the value that will be plotted for each slice in the doughnut graph. This list contains the captions of all fields returned by the source query that store numeric values *except* for the field selected in the **Category Column** list (if the field selected in this list stores numeric values). For example, the graph in the image that follows this list shows an example of a **Chart - Category** section whose graph plots the following values for the following slices in the graph:

  - High: 15

  - Medium: 9

  - Low: 1

  - Incomplete: 254
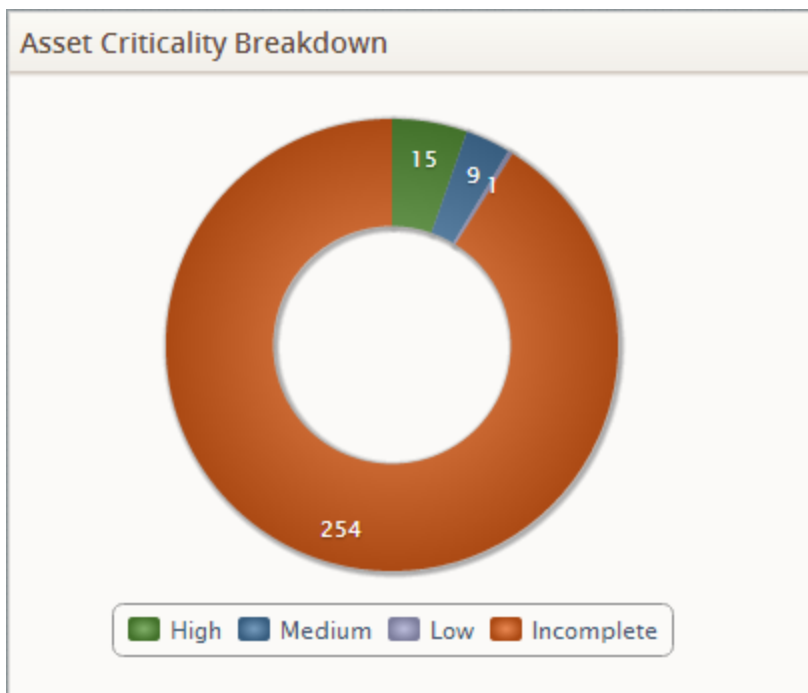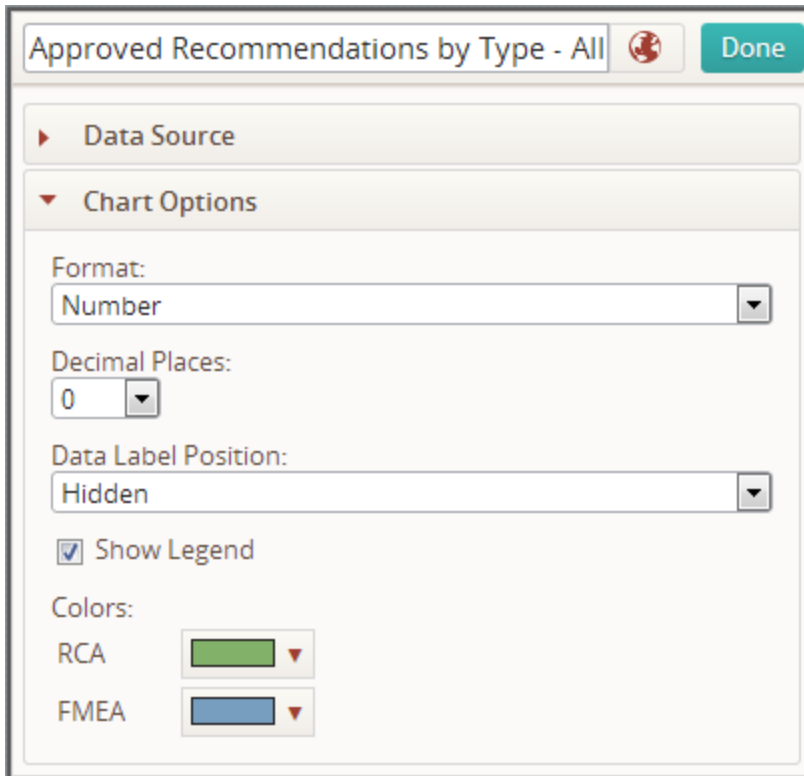
Chart Options Sub-Section

The **Chart Options** sub-section is a collapsible sub-section that provides options for spe-cifying how the doughnut graph is displayed in the **Chart - Category section** on the dash-board.

The following image shows an example of the **Chart Options** sub-section.

The **Chart Options** sub-section contains the following items that are used to configure the doughnut graph that is displayed in the **Chart - Series** section:

- **Format list:** Specifies the type of value that will be plotted in the doughnut graph. This list contains the following values:
    - **Currency:** Specifies that the values plotted on the graph are associated with numeric values that represent currency.
    - **Number:** Specifies that the values plotted on the graph are associated with numeric values. This value is selected by default.
    - **Percentage:** Specifies that the values plotted on the graph are associated with numeric values that represent a percentage.
- **Decimal places list:** Specifies the number of decimal places that will be displayed in the doughnut graph when you have specified that data labels should displayed on the graph. This list contains the following values that correspond with the num-ber of decimal places that will be displayed: *0, 1, 2, 3,* and *4.*

- **Data Label Position list:** Specifies whether or not data labels will be displayed in the doughnut graph.
- **Show Legend check box:** Specifies whether or not a legend is displayed for the graph. When this check box is:
  - *Selected*, a legend *is* displayed.
  - *Cleared*, a legend is *not* displayed.

  This check box is selected by default.

Data Label Position List

You can use the **Data Label Position** list in the **Chart Options** sub-section to specify whether or not data labels will be displayed in the doughnut graph. This list contains the following values:

- **Hidden:** Specifies that data labels should not be displayed on the graph, as shown in the following image.
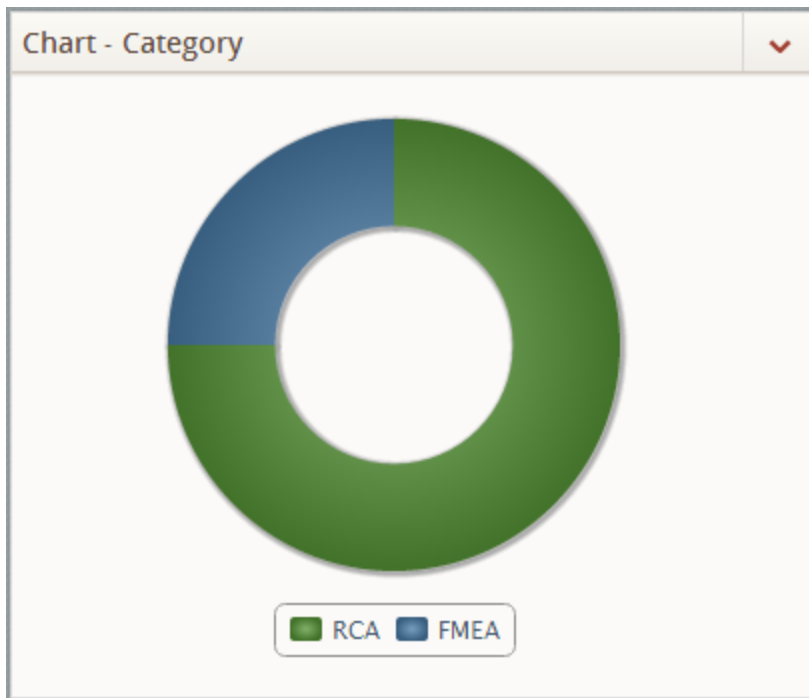
- **Inside Chart:** Specifies that data labels should be displayed on top of the graph, as shown in the following image.
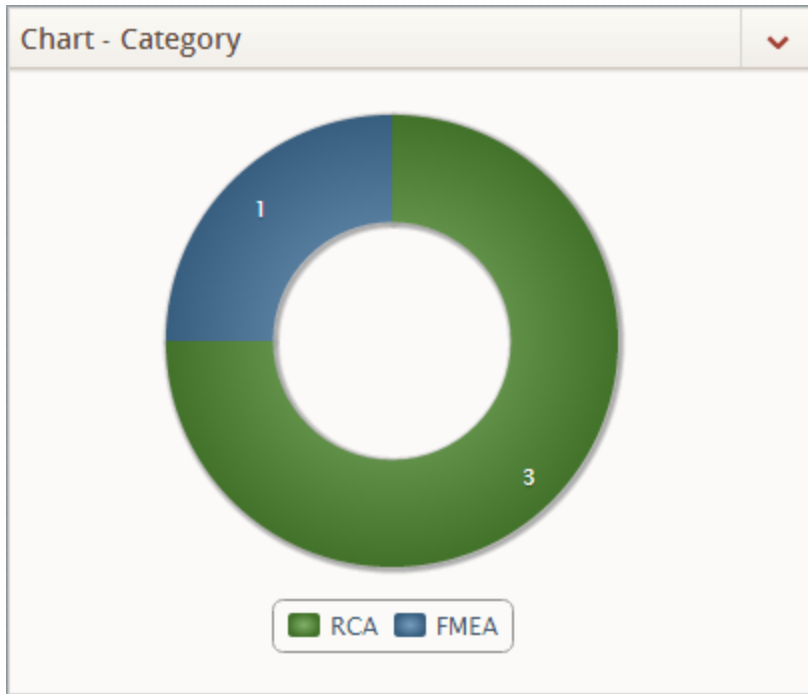
This value is selected by default.

- **Outside Chart:** Specifies that data labels should be displayed outside of the graph, as shown in the following image.
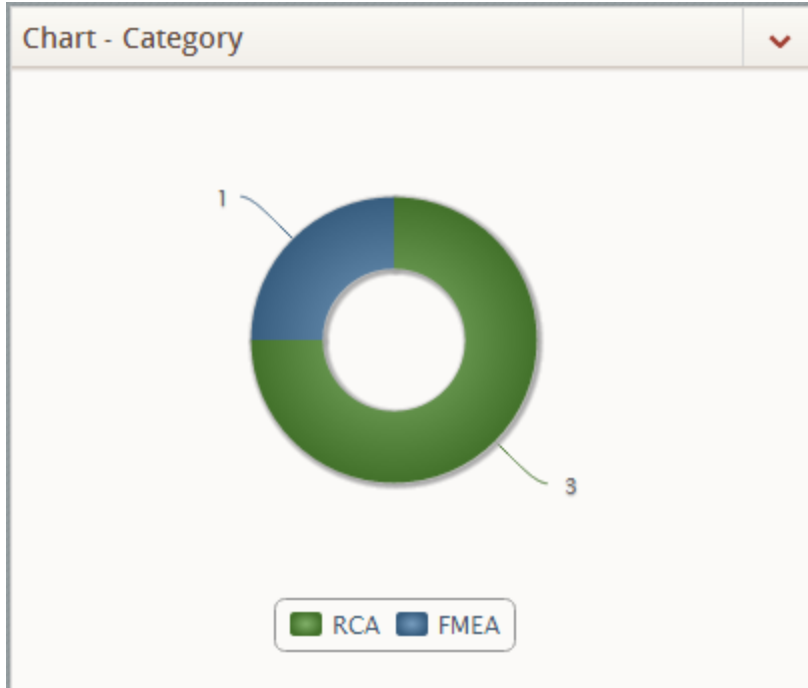
When you select this option, the doughnut graph is displayed at a smaller size automatically, as shown in the preceding image.

Color Boxes

The **Color** boxes in the [Chart Options sub-section](#) are used to specify the colors that will be used to plot each slice in the doughnut graph displayed in that section. Although the label for these boxes in the user interface is *Colors*, throughout the documentation we refer these as the *Color boxes.*

The number of boxes that are displayed and their labels correspond with the number of slices plotted on the graph (i.e., the number of unique values returned by the source query in the field that correspond with the category plotted in the graph, as specified in the **Category Column** list in the [Data Source sub-section](#)). For example, consider the following graph that plots the number and type of Recommendation records the Approved state.
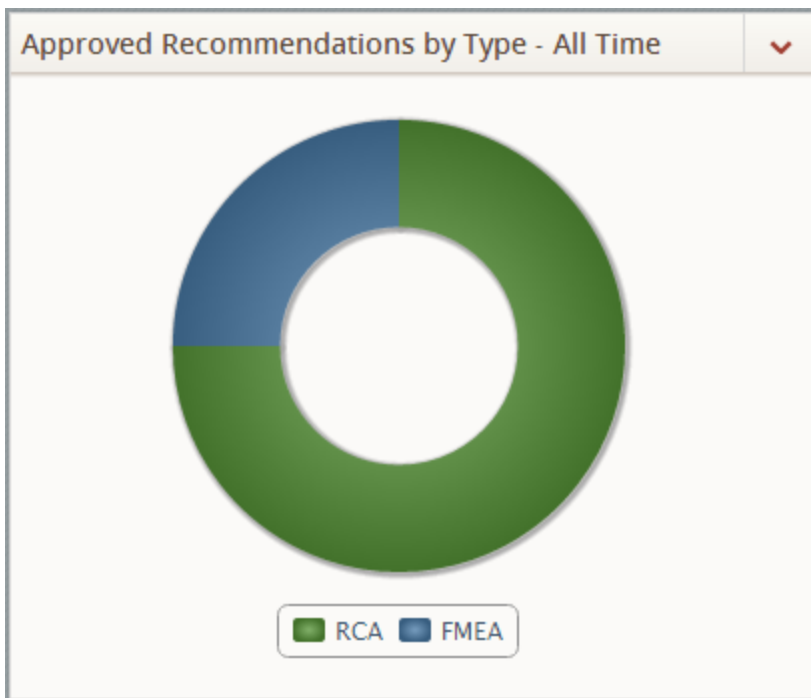
Continuing with the example, the Recommendation Type field returned by the source query has been specified as the category plotted on the graph, as shown in the following image. A red outline has been added to the image to highlight the **Category Column** list.

Color Boxes

As a result, one **Color** box is displayed for each unique value returned by the source query in the Recommendation Type field (i.e., the number of slices plotted on the graph). A red outline has been added to the following image to highlight these boxes.

The color that you select for each box will be used for the associated bar in the graph. When you click in these boxes, two color palettes are displayed:

- The first palette contains three colors that correspond with Health Indicator record statuses. A red outline has been added to the following image to highlight the first palette.

- The second palette contains 35 colors that do *not* correspond with Health Indicator record statues. A red outline has been added to the following image to highlight the second palette.

A white or black check mark appears on top of the color that is selected in the palette, as shown in the preceding images of the palettes.
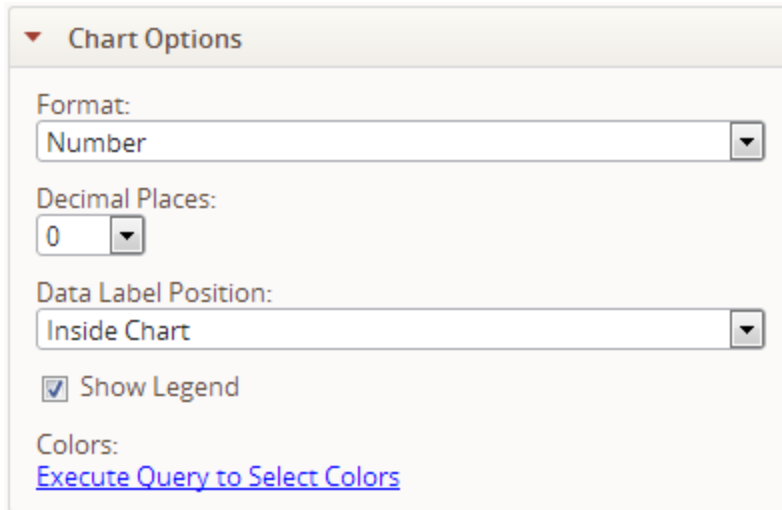
When you add a **Chart - Category** section to a dashboard and specify the category that should be plotted in the doughnut chart, the Meridium APM system automatically selects colors for the slices in the graph from the second palette, from left to right, top to bottom. For example, numbers have been added to the following image of the second palette to illustrate the order in which colors will be selected by the Meridium APM system.

You can modify the selection for any bar or line by clicking another color in the box.

When you are adding a new **Chart - Category** section to a dashboard and you access the **Chart Options** sub-section for the first time, the **Color** boxes are not displayed, and the text *Execute Query to Select Colors* appears as a hyperlink below the **Color** label, as shown in the following image.

You must click this link in order for the **Color** boxes to appear for each slice that will be plotted on the graph.

About the Chart - Series Section

A **Chart - Series** section on a dashboard displays a graphical representation of query results, as specified in the [edit view of the Chart - Series section](). Specifically, a **Chart - Series** section displays bar charts and line charts.

The following image shows an example of a **Chart - Series** section that has been configured for a dashboard.

About the Edit View of the Chart - Series Section

In addition to the label text box and Done button, the edit view of the **Chart - Series** section displays the following sub-sections that are used to configure the content that is displayed in a **Chart - Series** section on a dashboard:

- **Data Source** sub-section
- **Chart Options** sub-section
- **X-Axis** sub-section
- **Y-Axis** sub-section

The following image shows an example of the edit view of a **Chart - Series** section that is configured for a dashboard.

Data Source Sub-Section

The **Data Source** sub-section is a collapsible sub-section that provides options for specifying the source query for values plotted on the graph in a [Chart - Series section](#) on the dashboard. When you access the edit view of the **Chart - Series** section, the **Data Source** sub-section is displayed by default.

The following image shows an example of the **Data Source** sub-section.

The **Data Source** sub-section contains the following items:

- **Query Path text box:** Contains the Catalog location of the query whose results will be plotted in the graph in the [Chart - Series section](#) (i.e., the source query). To the right of this text box, the following buttons are displayed:

  - [...] : Displays the **Choose Query** dialog box where you can select the source query in the Meridium APM Catalog. When you specify a value in this text box using this button, if the selected query contains prompts, fields that correspond with these prompts appear in the **Query Parameters** pane.

  - [↻] : Refreshes the source query. If you type a value in the **Query Path** text box manually and the specified query contains prompts, you must click this button in order for the prompt fields to appear in the **Query Parameters** pane.

  If this text box does not contain a valid query path, a [validation message appears](#) and a red outline appears around this text box and the **Query Parameters** pane until you specify a valid query path.

- **Query Parameters pane:** If the source query contains prompts, this pane displays text boxes and lists where you should specify values for these prompts. The labels of the text boxes and lists correspond with the captions of prompt fields. If this pane contains lists and a value is not selected in each list, a [validation message appears](#) and a red outline appears around this text box and the **Query Path** text box until you select a value in each list.

In the image in this topic, the following items are displayed in the **Query Parameters** pane:

- **Route ID** list
- **Asset ID** list
- **Start Date** text box
- **End Date** text box

X-Axis Sub-Section

The **X-Axis** sub-section is a collapsible sub-section that displays options for configuring the x-axis of the graph that is displayed in the **Chart-Series** section on a dashboard.

The following image shows an example of the **X-Axis** sub-section.

The **X-Axis** sub-section contains the following items that are used to configure the graph that is displayed in the **Chart - Series** section:

- **X-Axis Label text box:** Specifies the label for the x-axis on the graph. In the image in this topic, this text box contains the value *Reading Taken Date*.

- **Minimum text box:** Specifies the smallest value plotted on the x-axis on the graph. You can type only numeric values in this text box. In the image in this topic, this text box contains the value *0.00* (zero).

- **Maximum text box:** Specifies the largest value plotted on the x-axis on the graph. You can type only numeric values in this text box. In the image in this topic, this text box contains the value *1.00*.

- **Hide Scale Labels check box:** Specifies whether or not labels will be displayed on the x-axis scale on the graph. This check box is cleared by default. When this check box is:

  - **Selected:** Labels will not be displayed on the x-axis scale. The following image shows an example of a graph in a **Chart Series** section when this check box is *selected*
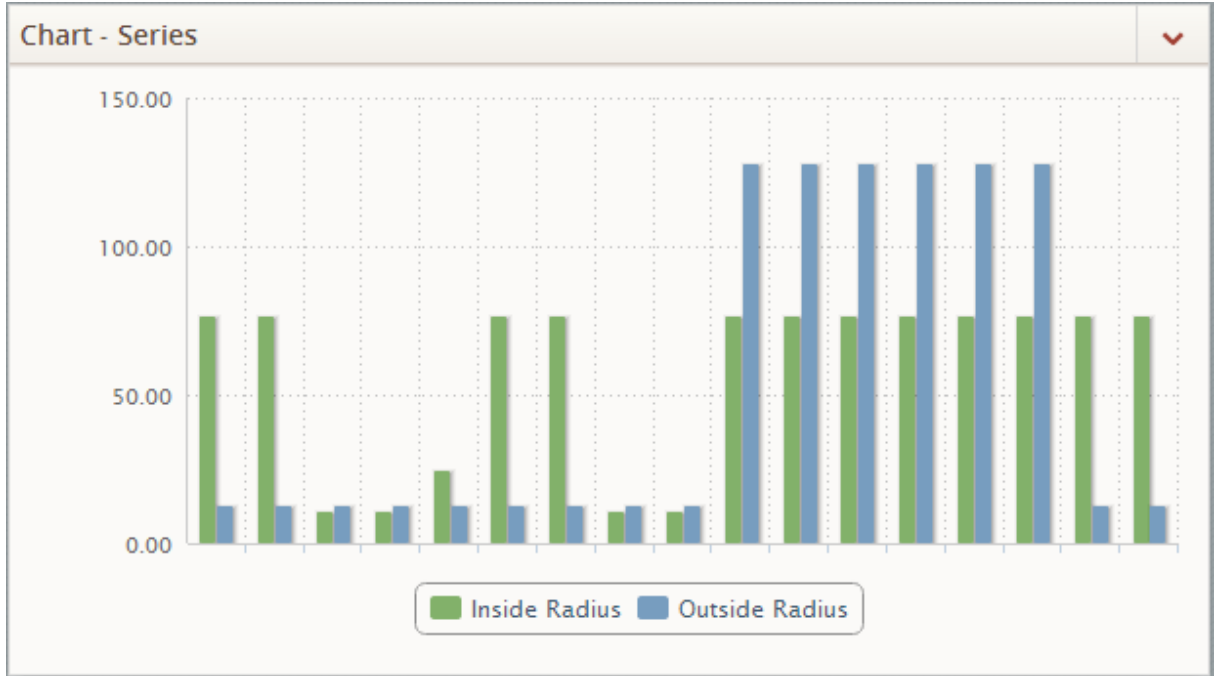
- **Cleared:** Labels will be displayed on the x-axis scale. The following image shows an example of a graph in a **Chart - Series** section when this check box is*cleared*.
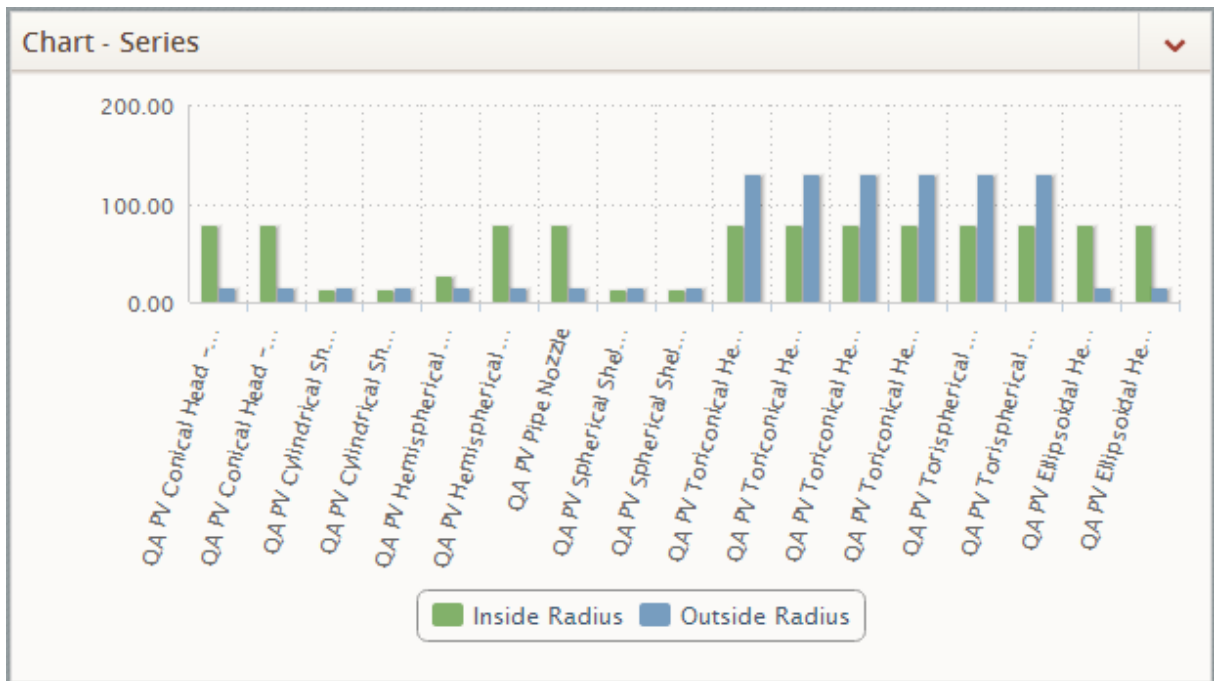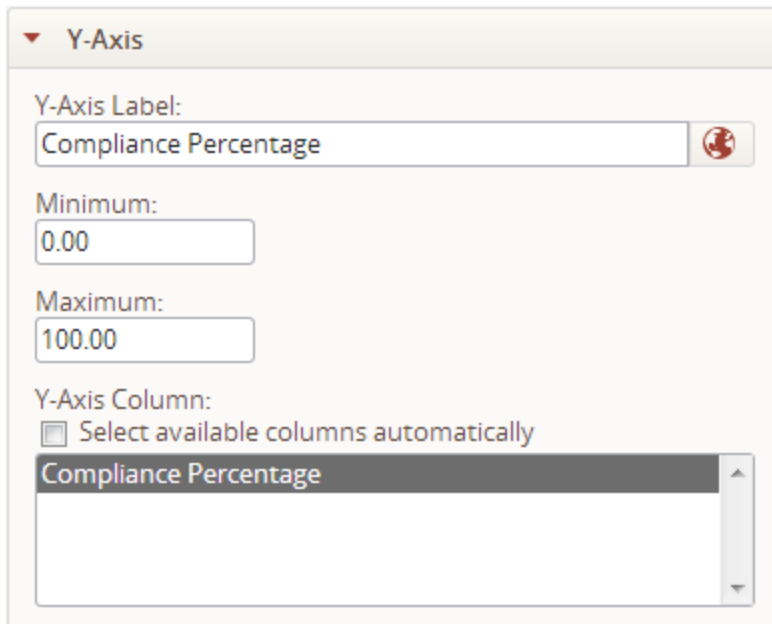
- **X-Axis Column list:** Specifies the field whose values are plotted on the x-axis on the graph. The values in this list correspond with the captions of fields in records that are returned by the underlying query for the dashboard section. The field caption associated with the first field displayed in the query results is selected by default. In the image in this topic, the value*Alert Level*is selected in this list.

Y-Axis Sub-Section

The **Y-Axis** sub-section is a collapsible sub-section that displays options for configuring the y-axis of the graph that is displayed in a **Chart-Series** section on a dashboard.

The following image shows an example of the **Y-Axis** sub-section.

The **Y-Axis** sub-section contains the following items that are used to configure the graph that is displayed in the **Chart - Series** section:

- **Y-Axis Label text box:** Specifies the label for the y-axis on the graph. In the image in this topic, this text box contains the value *Compliance Percentage*.

- **Minimum text box:** Specifies the smallest value plotted on the y-axis on the graph. You can type only numeric values in this text box. In the image in this topic, this text box contains the value *0.00* (zero).

- **Maximum text box:** Specifies the largest value plotted on the y-axis on the graph. You can type only numeric values in this text box. In the image in this topic, this text box contains the value *100.00*.

- **Y-Axis Column list:** Specifies the field whose values are plotted on the y-axis on the graph. The values in this list correspond with the captions of fields in records that are returned by the underlying query for the dashboard section. You can select multiple values in this list.

  Above this list, the **Select available columns automatically** check box is displayed. You can select this check box to specify that *all* values in the field identified in the **Y-Axis Column** list that correspond with numeric fields in records returned by the underlying query, with the exception of the value that is plotted on the x-axis, should be selected automatically, (i.e., plotted on the y-axis). This is useful when

you have specified a Crosstab query as the underlying query for the dashboard section.

For example, suppose that your dashboard contains a **Chart - Series** section whose underlying query is a Crosstab query whose results shows the number of equipment, by manufacturer, associated with the statuses of active, inactive, and unknown, as shown in the following table.

Manufacturer	Number of Active Equipment	Number of Inactive Equipment	Unknown
BALDOR	6	1	1
CENTURY	1	0	0
TOSHIBA	11	0	0

Continuing with the example, suppose that the manufacturer is plotted on the x-axis and the **Select available columns automatically** check box is selected. The number of equipment associated with the statuses of active, inactive, and unknown will be plotted on the y-axis of the graph, as shown in the following image, because they correspond with numeric fields returned by the query that are not plotted on the x-axis.
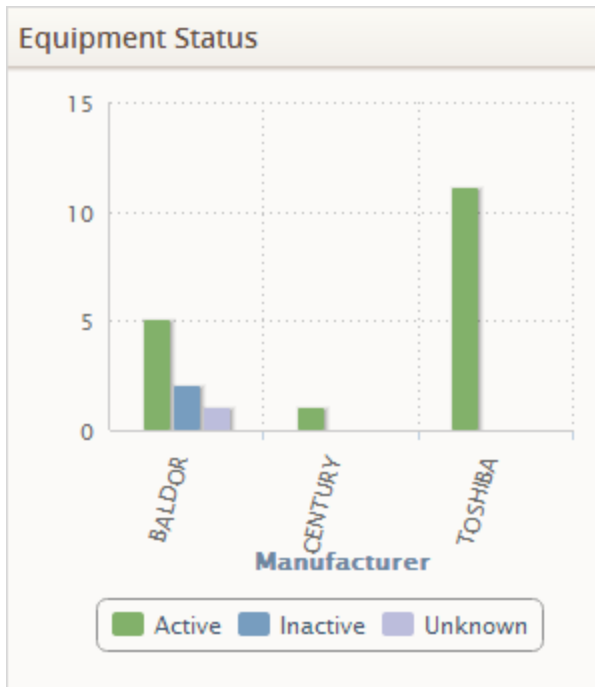
Continuing with the example, suppose that the results of this query change to reflect the following values.

Manufacturer	Number of Active Equipment	Number of Inactive Equipment
BALDOR	6	2
CENTURY	1	0
TOSHIBA	11	0

In other words, no equipment associated with an unknown status was returned by the query. As such, the next time you access the dashboard or if you refresh the dashboard, the y-axis will be updated automatically to plot only the number of active and inactive equipment, as shown in the following image.
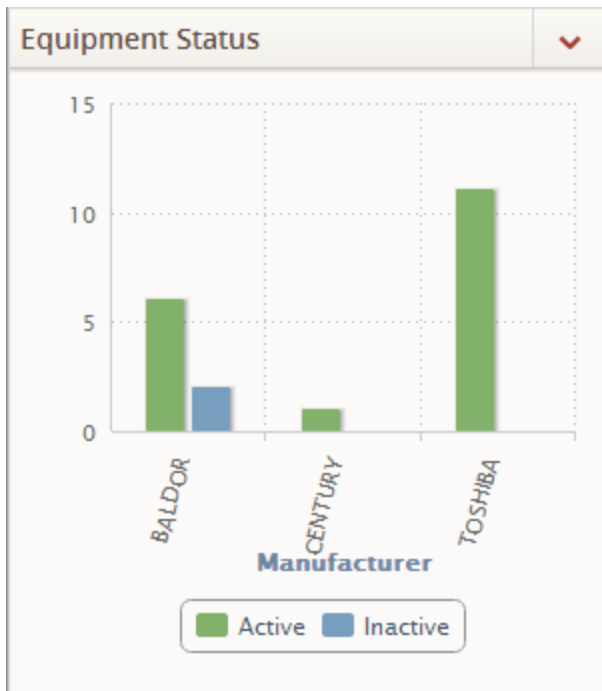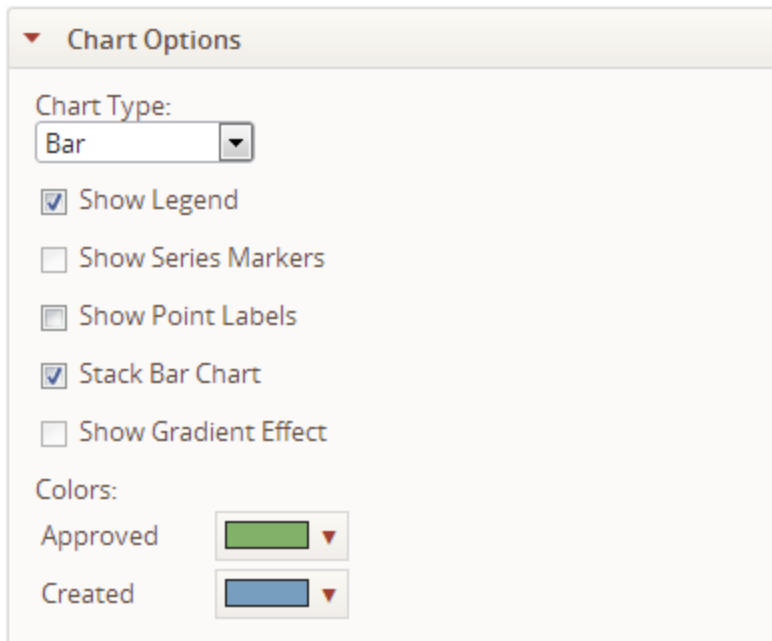
Chart Options Sub-Section

The **Chart Options** sub-section is a collapsible sub-section that provides options for specifying the type of graph and information that is displayed in the **Chart - Series** section on the dashboard.

The following image shows an example of the **Chart Options** sub-section.

The **Chart Options** sub-section contains the following items that are used to configure the graph that is displayed in the **Chart - Series** section:

- **Chart Type list:** Specifies the type of graph that is displayed in the section. This list contains the values *Line, Bar,* and *Curve.* The value *Line* is selected by default.

- **Show Legend check box:** Specifies whether or not as legend is displayed on the graph. When this check box is selected, a legend is displayed on the graph. This check box is *selected* by default.

- **Show Series Markers check box:** Specifies whether or not data points appear on the graph. When this check box is selected, data points appear on the graph. This check box is *cleared* by default, and it is enabled only when the value *Line* is selected in the **Chart Type** list.

- **Show Point Labels check box:** Specifies whether or not values are displayed above each data point that appears on the graph. When this check box is selected, values are displayed above each data point that appears on the graph. This check box is *cleared* by default.

- **Stack Bar Chart:** [Specifies whether or not matching, x-axis data point markers will be combined and displayed on top of one another](#).

- **Show Gradient Effect:** Specifies whether or not shading will be displayed below the highest point (i.e., largest value) plotted in the line graph. This check box is *cleared* by default, and it is enabled only when the value *Line* is selected in the **Chart Type** list.
- **Color boxes:** Specifies the colors that will be used to plot each bar or line on the graph.

Stack Bar Chart Check Box

The **Stack Bar Chart** check box in the [Chart Options sub-section](#) specifies whether or not matching, x-axis data point markers will be combined and displayed on top of one another. The **Stacked Bar Chart** check box is cleared by default and is enabled *only* when the value *Bar* is selected in the **Chart Type** list.

When this check box is:

- **Selected:** For bar graphs for which two or more columns or fields are measured on the y-axis, x-axis datapoint markers whose values are the *same* are combined and displayed on top of one another.

  For example, the following image shows an example of a graph in a **Chart - Series** section when the **Stack Bar Chart** check box is *selected*.
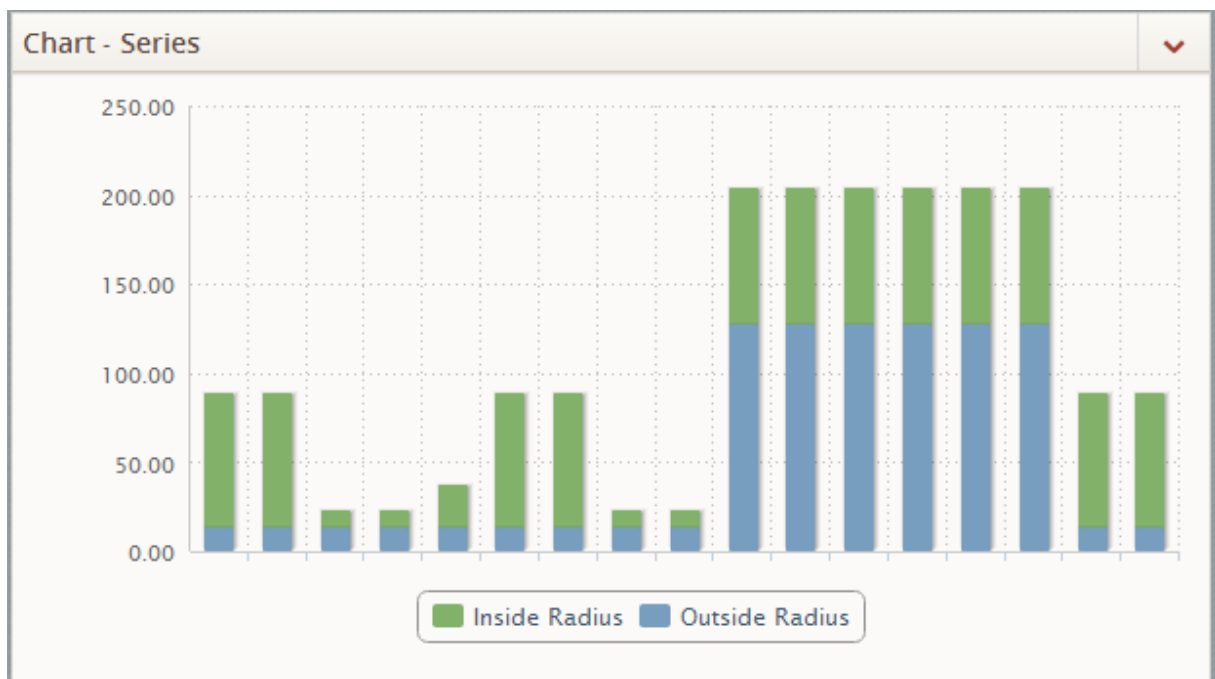
- **Cleared:** For bar graphs for which two or more columns or fields are measured on the y-axis. x-axis datapoint markers whose values are the *same* are *not* combined and displayed on top of another.

  Continuing with the example, the following image shows an example of the same graph when the **Stack Bar Chart** check box is *cleared*.

Stack Bar Chart Check Box
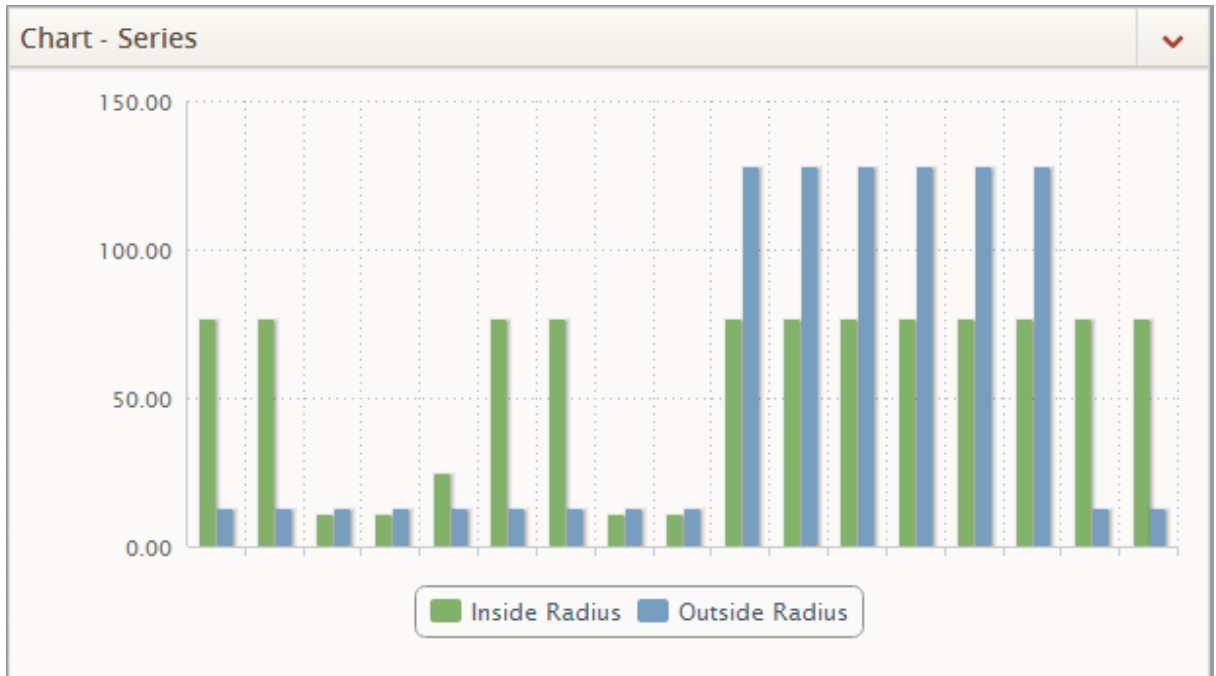
Show Gradient Effect Check Box

The **Show Gradient Effect** check box in the [Chart Options sub-section](#) specifies whether or not shading will be displayed below the highest point (i.e., largest value) plotted in the line graph. This check box is cleared by default and is enabled *only* when the value *Line* is selected in the **Chart Type** list. When this check box is:

- **Selected:** Shading is displayed below the highest point plotted in the line graph, as shown in the following image.
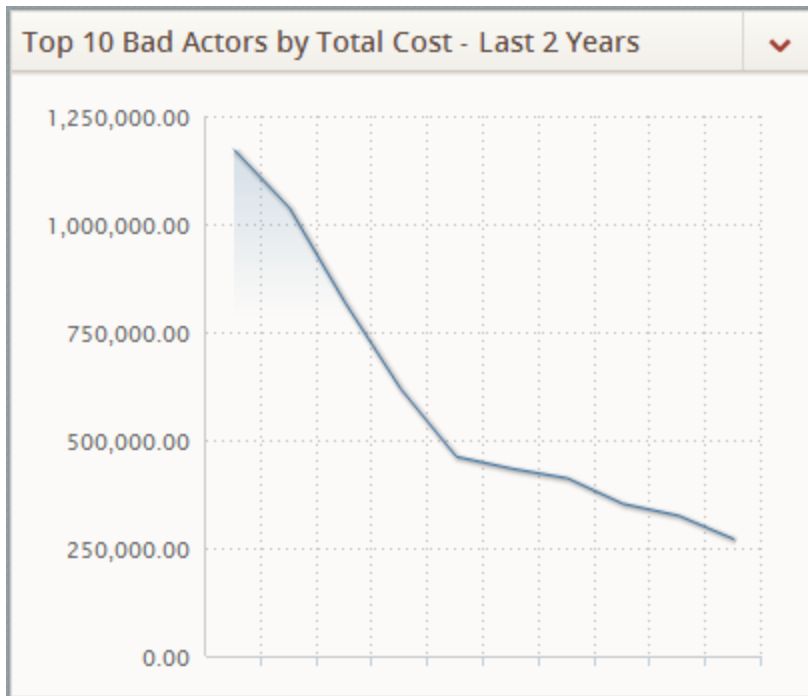
- **Cleared:** Shading is *not* displayed below the highest point plotted in the line graph, as shown in the following image.

Show Gradient Effect Check Box
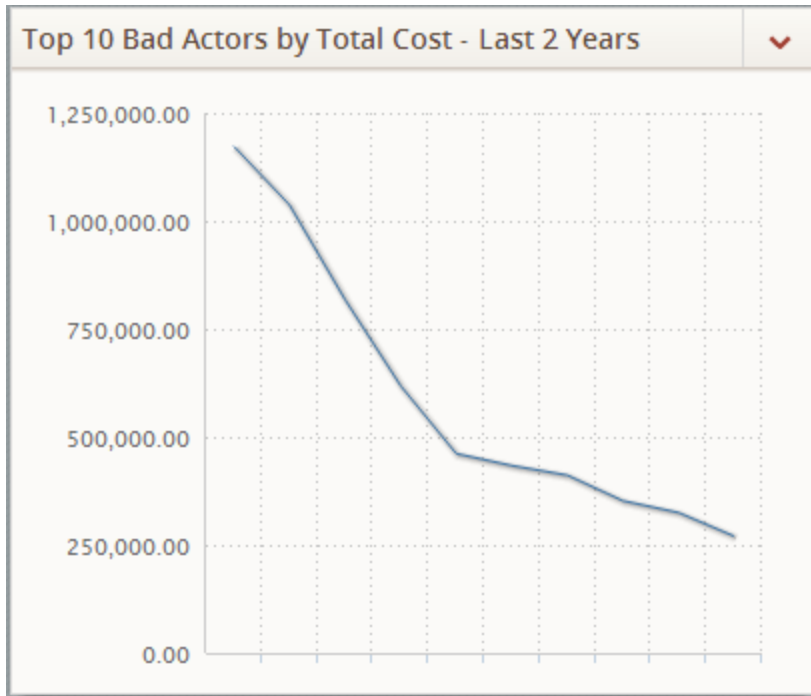
Color Boxes
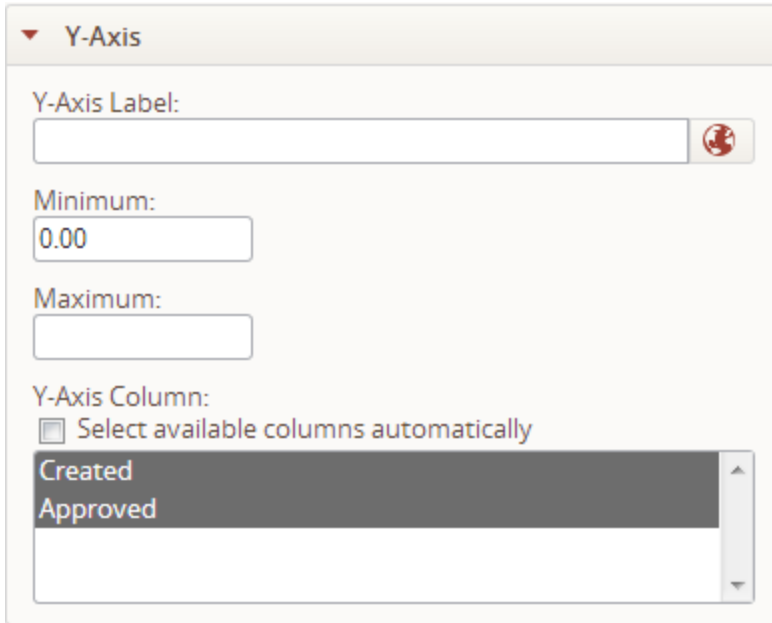
The **Color** boxes in the Chart Options sub-section are used to specify the colors that will be used to plot each bar or line in the graph displayed in that section. Although the label for these boxes in the user interface is *Colors*, throughout the documentation we refer these as the *Color* boxes.

The number of boxes that are displayed and their labels correspond with the number of fields whose values are plotted on the y-axis of the graph (specified in **Y-Axis Column** list in the Y-Axis sub-section) and their field captions. For example, consider the following section that plots the number of Recommendation records in the Created and Approved states over the past 12 months.

Continuing with the example, the **Y-Axis** sub-section in the edit view of this section specifies that two values are plotted on the y-axis, Recommendation records in the Created state and Recommendation records in the Approved state, as shown in the following image.
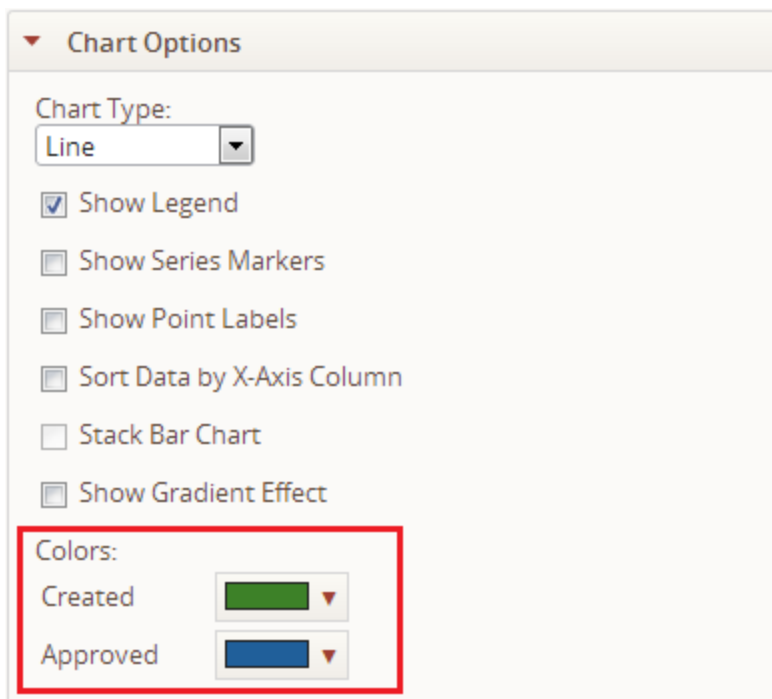
Color Boxes

As a result, one **Color** box is displayed for each of these values in the **Chart - Options** sub-section. A red outline has been added to the following image to highlight these boxes.

The color that you select for each box will be used for the associated bar in the graph. When you click in these boxes, two color palettes are displayed:

- The first palette contains three colors that correspond with Health Indicator record statuses. A red outline has been added to the following image to highlight the first palette.

- The second palette contains 35 colors that do *not* correspond with Health Indicator record statues. One of the following second palettes is displayed, depending upon the graph type (bar or line). Red outlines have been added to the following images to highlight the second palette.

    - For line graphs, the following palette is displayed.

    - For bar graphs, the following palette is displayed.

A white or black check mark appears on top of the color that is selected in the palette, as shown in the preceding images where the second palette is outlined in red.

When you add a **Chart - Series** section to a dashboard and specify fields whose values should be plotted on the y-axis, the Meridium APM system automatically selects colors for the bars or lines in the graph from the second palette, from left to right, top to bottom. For example, numbers have been added to the following image of one of the second palettes to illustrate the order in which colors will be selected by the Meridium APM system.

This same selection concept applies for both variations of the second palette. You can modify the selection for any bar or line by clicking another color in the box.

When you access the **Chart Options** sub-section for a **Chart - Series** section for which you have not specified a source query or the values that should be plotted on the y-axis, the **Color** boxes are not displayed and the text *No Columns Selected* appears below the **Color** label, as shown in the following image.
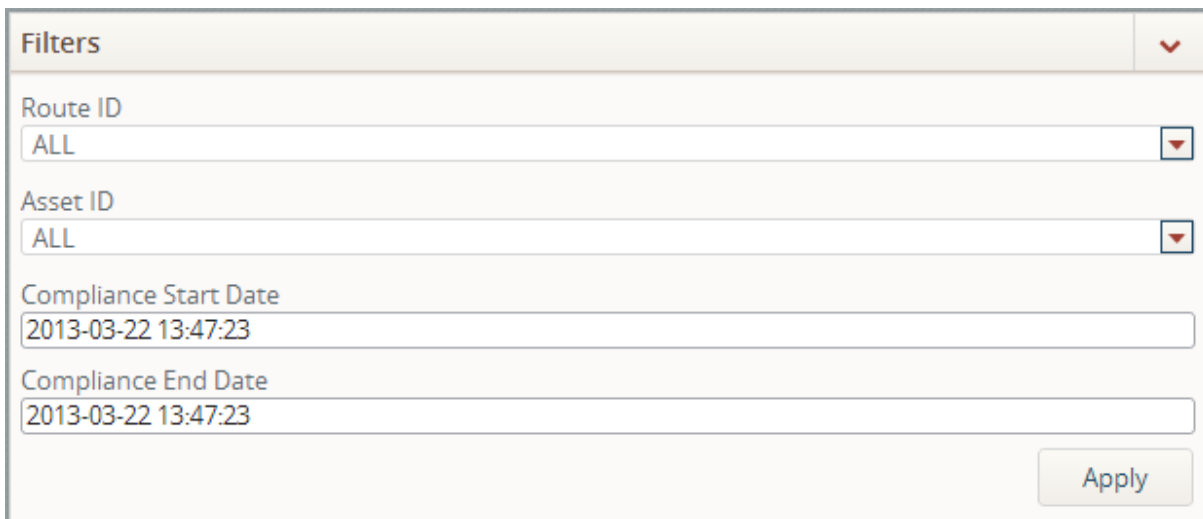
About the Content Filter Section

You can use a **Content-Filter** section on a dashboard to filter the content that is displayed in all the other types of dashboard sections *except* the following:

- Hyperlinks
- Measures - Linked

A **Content Filter** section displays text boxes and lists that correspond with the captions of prompt fields in the query that is specified in the edit view of the **Content Filter** section. The following image shows an example of a **Content Filter** section that has been configured for a dashboard.

The Content Filter section contains the following items:

- **Text boxes and/or lists:** One text box or list is displayed for each prompt field in the query that is specified in the edit view of the **Content Filter** section. Each list is populated with values that correspond with records that exist in your database. When you select values in these lists and type values in these text boxes, these values are passed into the corresponding prompts in other sections on the dashboard, filtering the content that is displayed in these sections. Each text box and list must contain a value in order to apply the filter to the dashboard. Throughout the documentation, the term *filter* is used to refer to these text boxes and lists and the values that you have specified for them.
- **Apply button:** Used to apply the filter to the applicable dashboard sections.

For example, the underlying queries to the following sections on the on the Operator Rounds dashboard contain prompts on the following fields in the following families.

This dashboard section's underlying query contains prompts on...	...these fields...	...in these families.
Active Recommendations	Route ID Asset ID	Route Operator Rounds Recommendation
Health Indicators	Route ID Related Asset ID	Route Measurement Location
Overdue Measurement Locations	Route ID Related Asset ID	Route Measurement Location

As a result, when you specify values in the **Route ID** and **Asset ID** lists in the **Filters** section on the Operator Rounds dashboard, (a configured **Content Filter** section) these values are passed into the corresponding prompts contained in the underlying queries in the sections listed in the table, determining the content that is displayed in these sections.

Consider the following image that shows an example of the Operator Rounds dashboard when the following values and selections exist for the following items in the **Filter** section:

- **Route ID list:** *ALL*
- **Asset ID list:** *ALL*
- **Compliance Start Date text box:** *02/17/2013 15:04:44*
- **Compliance End Date text box:** *04/19/2013 15:04:44*

Continuing with the example, now consider the following image that shows an example of the Operator Rounds dashboard when the values selected in the **Route ID** and **Asset ID** lists are changed from *ALL* to *10 R* and *PMP-3830* respectively. When you apply the filter (click the **Apply** button), the content on in these sections is filtered according to your selections in these lists.

About the Edit View of the Content Filter Section

The edit view of the **Content Filter** section contains fields that are used to specify the name of the **Content Filter** section that is displayed on the dashboard, and the query whose prompt fields will be displayed in the section.

The following image shows an example of the edit view of a **Content Filter** section that has been configured for the baseline Operator Rounds dashboard.
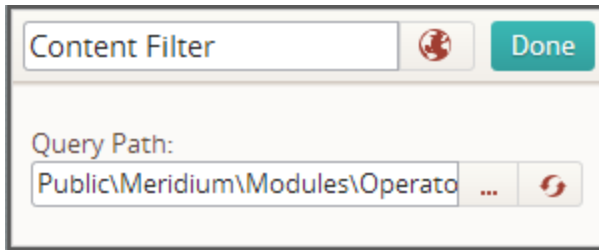
In addition to the label text box and **Done** button, the edit view of the **Content Filter** section contains the **Query Path** text box, which displays the Catalog location of the query whose prompt fields will be used to filter the content that is displayed on the dashboard. To the right of this text box, the following buttons are displayed:

- **...** : Displays the **Choose Query** dialog box, where you can select a query in the Catalog.

- ↻ : Refreshes the query associated with the Catalog item specified in the **Query Path** text box. If you type a value in the **Query Path** text box manually, you must click this button in order for the prompt fields to appear in the **Query Parameters** pane.

If this text box does not contain a valid query path, a validation message appears and a red outline appears around this text box until you specify a valid query path.

Hyperlinks Section
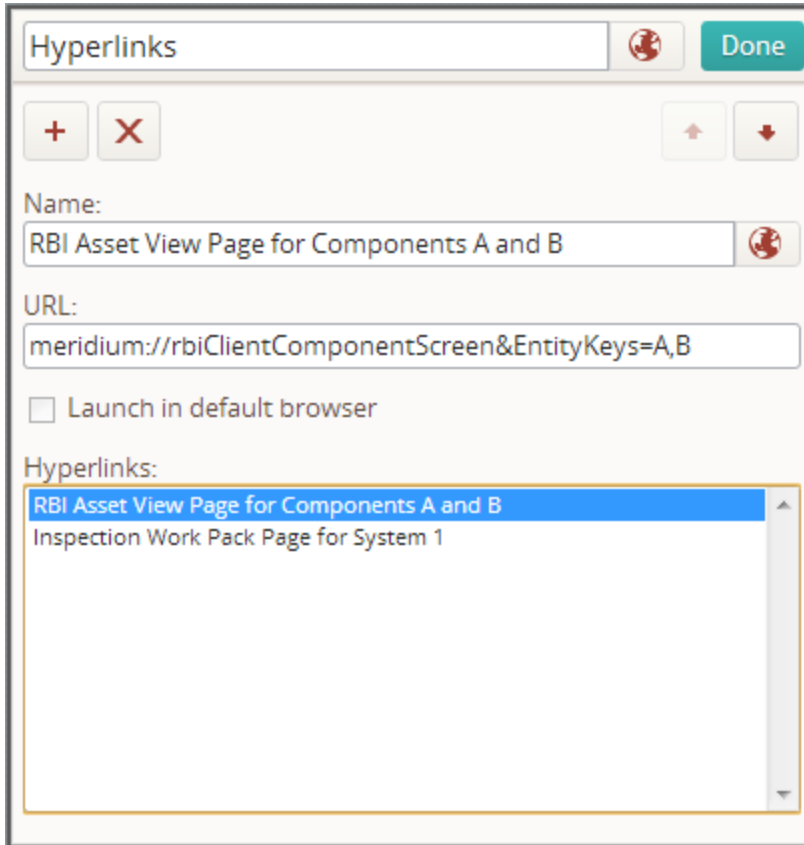
A **Hyperlinks** section on a dashboard displays hyperlinks to URLs, as specified in the [edit view of the **Hyperlinks** section]().

The following image shows an example of a **Hyperlinks** section containing two hyperlinks that provide navigation to two different pages in the Meridium APM Framework.
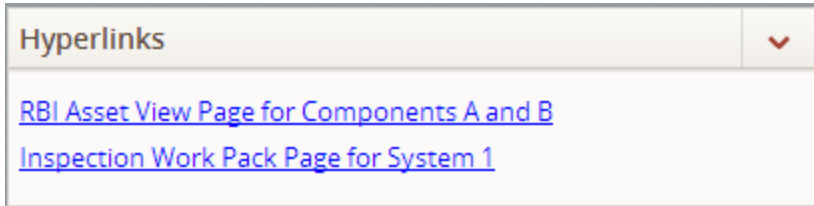
About the Edit View of the Hyperlinks Section

The edit view of the **Hyperlinks** section provides options that are used to configure the content that is displayed in the **Hyperlinks section** on the dashboard.

In addition to the label text box and **Done** button, the edit view of the **Hyperlinks** section contains the following items:

- ![+ button] **button:** Enables the **Name** and **URL** text boxes, and displays a blank entry in the **Hyperlinks** list.

- ![X button] **button:** After displaying a confirmation message, removes the selected hyperlink from the **Hyperlinks** list.

- **Name text box:** Specifies the label for the hyperlink. When you access the edit view of the **Hyperlinks** section for the first time or when no hyperlinks are defined, this text box is disabled. You must click the ![+ button] button in order to enable this text box.

- **URL text box:** Specifies the URL that is associated with the hyperlink. When you access the edit view of the **Hyperlinks** section for the first time or when no

hyperlinks are defined, this text box is disabled. You must click the [+] button in order to enable this text box. When this text box is enabled, it contains the value *meridium://*, the scheme for internal Meridium APM URLs, by default. You can create links that [use certain other schemes](#).

- **Launch in default browser check box:** Specifies whether or not the associated URL will be launched in your default web browser application. This check box is enabled only when the URL specified in the **URL** text box contains one of the following schemes:

  - http://
  - https://

  When this check box is:

  - *Selected*, the associated URL will be launched in your default web browser application.
  - *Cleared,* the associated URL will be launched in the Meridium APM Framework application.

- **Hyperlinks list:** Displays a list of the hyperlinks that you have defined. The order in which hyperlinks appear in this list reflects the order in which hyperlinks will appear in this section.

- [▲] **button:** Moves the selected value up in the **Hyperlinks** list. This button is enabled only when more than one value is displayed in the **Hyperlinks** list and when the selected value is not the first value in the list.

- [▼] **button:** Moves the selected value down in the **Hyperlinks** list. This button is enabled only when more than one value is displayed in the **Hyperlinks** list and when the selected value is not the last value displayed in the list.

Supported URL Schemes

The following table lists the URL schemes are supported for the [Hyperlinks section](#) and provides examples of URL syntaxes that use those schemes.

Scheme	Example URL(s)	Description

/Meridium/	• /Meridium/Dashboard/Index.html?path=<dashboard Catalog location>
	
	• /Meridium/Policies/<Page parameter>.html
		• This syntax can be used to create a link to another dashboard, where *<dashboard Catalog location>* is the Catalog location of the dashboard that you want to view on the **Dashboard** page.
		
		For example, the URL /Meridium/Dashboard/Index.html?path**=**Public\Meridium\Modules\Operator Rounds\Dashboards\OPR Dashboard will display the baseline Policy Manager dashboard on the **Dashboard** page.
		
		• This syntax can be used to create a link to a page in the Policy Manager module, where *<Page parameter>* accepts the following values:
		  • **Search:** Displays the **Policies Search** page.
		  • **Policy:** Displays the **Policy** page.
		
		For example, the URL /Meridium/Policies/Search.html will display the **Policies Search** page.

http://	http:<URL for external site>	This syntax can be used to create a link to a URL that is external to the Meridium APM Framework application, where *<URL to external site>* is, for example, the URL to your company's intranet site.

Scheme	Example URL(s)	Description
https://	https://<URL for external site>	This syntax can be used to create a link to a URL that is external to the Meridium APM Framework application, where *<URL to external site>* is, for example, the URL to your company's intranet site.

List - Query Result Section

The **List - Query Result** section displays a grid containing the results of the query specified in the edit view of the **List - Query Result** section.

The following image shows an example of a **List - Query Result** section that has been configured for the baseline Operator Rounds dashboard.
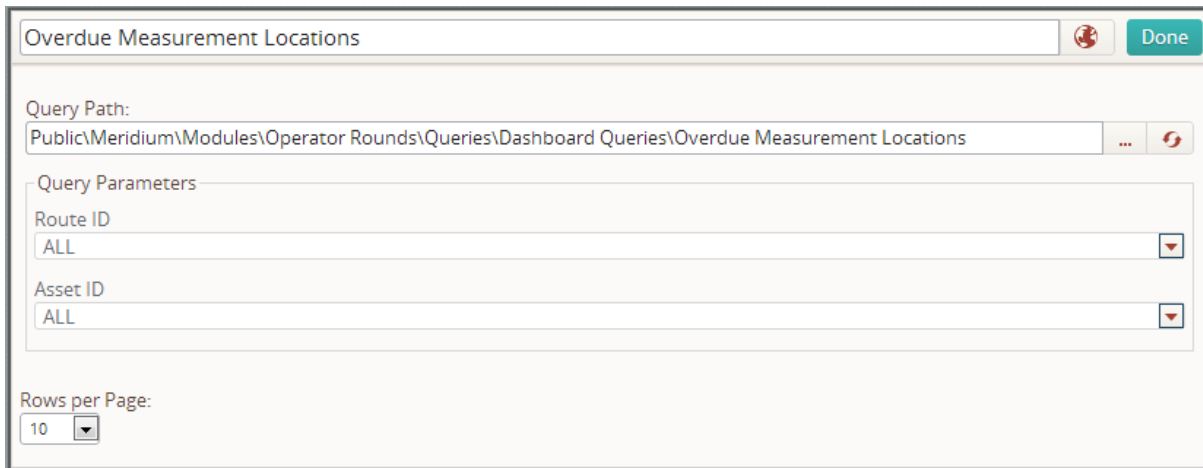
Overdue Measurement Locations			⌄
ML ID ⇕	**ML Description** ⇕	**Due Date** ⇕	
PMP-3827~Filter Check	PMP-3827	10/09/2007 11:00:00	
PMP-3822~Filter Check	PMP-3822	10/09/2007 11:00:00	
PMP-3820~Filter Check	PMP-3820	10/09/2007 11:00:00	
PMP-5211~Pump Stroke Check	General Check PMP-5211	10/09/2007 11:30:00	
PMP-3834~Pump Check	Pump Check PMP-3834	10/09/2007 11:30:00	
PMP-3902~Pump Stroke Check	Inspect Pump Stroke	10/09/2007 11:30:00	
PMP-3821~Filter Check	PMP-3821	10/09/2007 12:00:00	
PMP-3819~Filter Check	PMP-3819	10/09/2007 17:00:00	
PMP-3830~General Check	General Check PMP-3830	10/10/2007 06:30:00	
PMP-3830~Pump Stroke Check	General Check PMP-3830	10/10/2007 06:30:00	
Page 1 of 137 1,368 record(s)		⏮ ◀ ▶ ⏭	

About the Edit View of the List - Query Result Section

The edit view of the **List - Query Result** section provides options that are used to configure the content that is displayed in a **List - Query Result** section on a dashboard.

The following image shows an example of the edit view of a **List - Query** result section that has been named *Overdue Measurement Locations*.

In addition to the label text box and **Done** button, the edit view of the **List - Query Result** section contains the following items:

- **Query Path text box:** Defines the Catalog location of the query whose results will be displayed in the section (i.e., the source query). To the right of this text box, the following buttons are displayed:

    - **...** : Displays the **Choose Query** dialog box where you can select the query in the Meridium APM Catalog.

    - **⟳** : Refreshes the source query. If you type a value in the **Query Path** text box manually and the specified query contains prompts, you must click this button in order for the prompt fields to appear in the **Query Parameters** pane.

    If this text box does not contain a valid query path, a validation message appears and a red outline appears around this text box and the **Query Parameters** pane until you specify a valid query path.

- **Query Parameters pane:** If the source query contains prompts, this pane displays text boxes and lists where you should specify values for these prompts. The labels of the text boxes and lists correspond with captions of prompt fields. If this pane contains lists and a value is not selected in each list, a validation message appears and a red outline appears around this text box and the **Query Path** text box until you select a value in each list.

Note: If the source query contains a prompt that accepts more than one value, you will be able to select more than one value from the drop-down lists in the **Query Parameters** pane.

- **Rows per Page list:** Your selection in this list determines the number of rows that will be displayed on each page in the grid that is displayed in the dashboard section. This list contains values in five value increments from *5* to *50*. The value *10* is selected in this list by default.

List - Single Column Section

A **List - Single Column** section displays a list of values that are stored in one field in records returned by a query specified in the edit view of the **List - Single Column** section.

The following image shows an example of a configured **List - Single Column** section that is displayed on the baseline Mechanical Integrity dashboard.

Depending upon your selections in the edit view, you can click values displayed in this section to view additional details about the specific record or view information in another dashboard section for the selected record.

About the Edit View of the List - Single Column Section

The edit view of the **List - Single Column** section provides options that are used to configure the content that is displayed in a [List - Single Column section](#) on a dashboard.

The following image shows an example of the edit view of a **List - Query** result section that has been named *Equipment Outside Risk Policy*.

In addition to [the label text box and **Done** button](#), the edit view of the **List - Single Column** section contains the following items:

- **Query Path text box:** Defines the Catalog location of the query whose results will be displayed in the section (i.e., the source query). To the right of this text box, the following buttons are displayed:
    - ⬛ **...** : Displays the **Choose Query** dialog box where you can select the source query in the Meridium APM Catalog.
    - ⬛ 🔄 : Refreshes the source query associated. If you type a value in the **Query**

Path text box manually and the specified query contains prompts, you must click this button in order for the prompt fields to appear in the **Query Parameters** pane.

If this text box does not contain a valid query path, a validation message appears and a red outline appears around this text box and the **Query Parameters** pane until you specify a valid query path.

- **Query Parameters pane:** If the source query contains prompts, this pane displays text boxes and lists where you should specify values for these prompts. The labels of the text boxes and lists correspond with the captions of prompt fields. If this pane contains lists and a value is not selected in each list, a validation message appears and a red outline appears around this text box and the **Query Path** text box until you select a value in each list.

- **Items per Page list:** Determines the number of rows that will be displayed on each page in the section. This list contains values in five value increments from *5* to *50*. The value *10* is selected in this list by default.

- **Column to Display list:** Determines the field whose values will be displayed in the section. The values in this list correspond with the captions of fields returned by the source query. The first field returned by the query is selected in this list by default.

- **On Click pane:** Contains a list that displays the following values:

  - **View Detail:** If you select this value, when you click a value in this section on the dashboard, a detail dialog box will be displayed. This value is selected by default.

  - **Send messages to other widgets:** If you select this value, when you select a value in this section on the dashboard, the value will be highlighted in yellow, and depending upon your selections in the following lists that are displayed and your selections in the edit-view of the **Measures - Linked** section, the content displayed in the **Measures - Linked** section will be specific to the selected value:

    - **Message ID text box:** The value that you specify in this text box must exactly match the value in the **Message Channel** text box in the edit-view of the **Measures - Linked** section.

    - **Message Contents list:** Your selection in this list defines the field in the query specified in the **Query Path** text box whose value will be passed into the prompt contained in the underlying query in the **Measures - Linked** section in order to determine the content that is displayed in the **Measures - Linked** section. This list is populated with fields returned by the source query.

The **On Click** pane is enabled only when a source query is specified.

List - Summary Section

A **List - Summary** section displays a list of values that are stored in records returned by the query specified in the edit view of the **List - Summary** section. Specifically, one row is displayed in the list for each record returned by the query, and each row displays the values that are stored in up to three fields in that record.

The following image shows an example of a configured **List - Summary** section that is displayed on the baseline Mechanical Integrity dashboard. Callouts have been added to the image to identify, within each row, where values are displayed that correspond with your selections in lists in the edit view of the **List - Summary** section.

Callout	Displays values that are stored in the field selected in this list in the edit view of the List - Summary section
1	Header
2	Detail
3	Status

Depending upon your selections in the edit view, you can click values displayed in this section to view additional details about the specific record or view information in another dashboard section for the selected record.

About the Edit View of the List Summary Section

The edit view of the **List - Summary** section provides options that are used to configure the content that is displayed in a **List - Summary** section on a dashboard.

The following image shows an example of the edit view of a **List - Summary** section that has been named *Pending RBI Recommendations*.

In addition to the label text box and Done button, the edit view of the **List - Summary** section contains the following items:

- **Query Path text box:** Defines the Catalog location of the query whose results will be displayed in the section (i.e., the source query). To the right of this text box, the following buttons are displayed:

    - [...] : Displays the **Choose Query** dialog box where you can select the source query in the Meridium APM Catalog.

    - [🔄] : Refreshes the source query. If you type a value in the **Query Path** text box manually, and the specified query contains prompts, you must click this

button in order for the prompt fields to appear in the **Query Parameters** pane.

If this text box does not contain a valid query path, a [validation message appears](#) and a red outline appears around this text box and the **Query Parameters** pane until you specify a valid query path.

- **Query Parameters pane:** If the source query contains prompts, this pane displays text boxes and lists where you should specify values for these prompts. The labels of the text boxes and lists correspond with the captions of prompt fields. If this pane contains lists and a value is not selected in each list, a [validation message appears](#), and a red outline appears around this text box and the **Query Path** text box until you select a value in each list.

- **Items per Page list:** Your selection in this list determines the number of rows that will be displayed per page in the section. This list contains values in 5 value increments from *5* to *50*. The value *10* is selected in this list by default.

- **Header list:** This list is populated with the captions of fields returned by the query specified in the **Query Path** text box. For each record returned by this query, values that are stored in the field that you select in this list will be displayed in bold at the top left of each row in the **List - Summary** section. A red outline has been added to the following image of a row in the **List - Summary** section to show where values that correspond with your selection in this list are displayed.

- **Detail:** This list is populated with the captions of fields returned by the query specified in the **Query Path** text box. For each record returned by this query, values that are stored in the field that you select in this list will be displayed at the bottom left of each row in the **List - Summary** section. A red outline has been added to the following image of a row in the **List - Summary** section to show where values that correspond with your selection in this list are displayed.

- **Status list:** This list is populated with the captions of fields returned by the query specified in the **Query Path** text box. For each record returned by this query, values that are stored in the field that you select in this list will be displayed at the top right of each row in the **List - Summary** section. A red outline has been added to the following image of a row in the **List - Summary** section to show where values that correspond with your selection in this list are displayed.
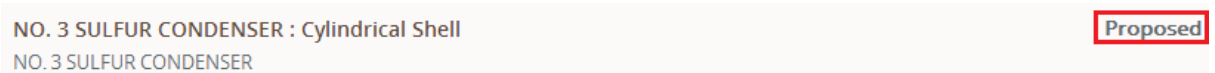
- **On Click pane:** Contains a list that displays the following values:
  - **View Detail:** If you select this value, when you click in this section on the dashboard, a [detail dialog box](#) will be displayed for the corresponding

record. This value is selected by default.

- **Send messages to other widgets:** If you select this value, when you select a value in this section on the dashboard, the value will be highlighted in yellow, and depending upon your selections in the following lists that are displayed and your selections in the edit-view of the Measures - Linked section , the content that is displayed in theMeasures - Linkedsection will be specific to the selected value:

    - **Message ID text box:** The value that you specify in this text box must exactly match the value in the **Message Channel** text box in the edit-view of the Measures - Linked section.

    - **Message Contents list:** Your selection in this list defines the field in the query specified in the **Query Path** text box whose value will be passed into the prompt contained in the underlying query in the **Measures - Linked** section in order to determine the content that is displayed in the Measures - Linked section. This list is populated with fields returned by the query specified in the **Query Path** text box.

The **On Click** pane is enabled only when a query is specified in the **Query Path** text box.

About the Measures - General Section

A **Measures - General** section on a dashboard displays content that is determined by various queries and options that are specified in the [edit view of the **Measures - General** section](). This content represents *measures* for equipment or locations that you analyze in Meridium APM. Throughout this section of the documentation, the term *measure* is used to refer to a classification of information that you want to analyze about a piece of equipment or a group of equipment.

The following image shows an example of a **Measures - General** section that has been named *General Measures*.
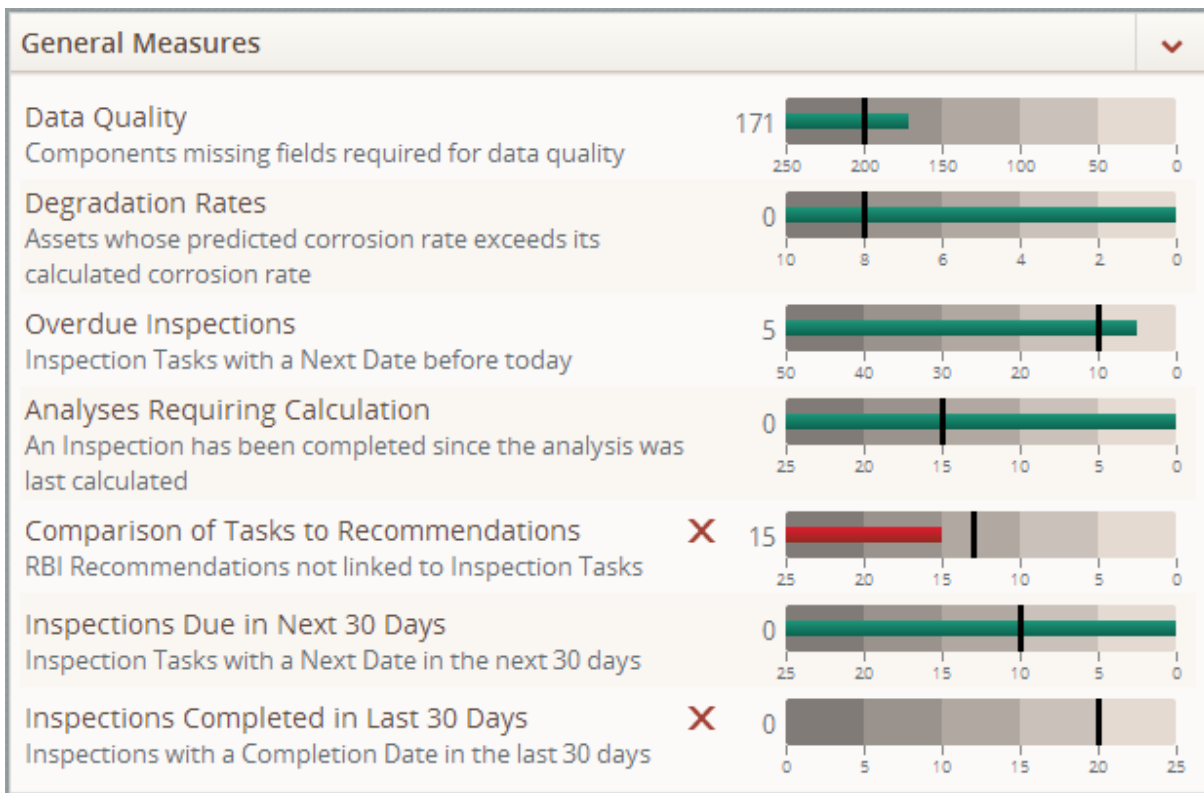
About the Edit View of the Measures - General Section

In addition to the label text box and **Done** button, the edit view of a **Measures - General** section contains the following sub-sections that are used to configure the content that is displayed in a **Measures - General** section on a dashboard:

- **Display Options** sub-section
- **Edit Measures** sub-section
- **Measure** sub-section
- **Data Source** sub-section
- **Scale** sub-section
- **Ranges** sub-section

The following image shows an example of the edit view of a **Measures - General** section that has been named *General Measures*.

General Measures		🌐	Done

Display Options

☑ Display Measure Description
☑ Indicate Current Value Meets Target
☑ Display Current Value As Graph
☑ Display Current Value

Edit Measures

+	**✕**		**⬆**	**⬇**

Data Quality
Degradation Rates
Overdue Inspections
Analyses Requiring Calculation
Comparison of Tasks to Recommendations
Inspections Due in Next 30 Days
Inspections Completed in Last 30 Days

▼ Measure

Name:
Data Quality 🌐

Description
Components missing fields required for data quality 🌐

Target:
200

▶ Data Source

▶ Scale

▶ Ranges

Display Options Sub-Section

The **Display Options** sub-section displays options that you can use to specify how information is displayed in the **Measures - General** section on the dashboard. The following check boxes are displayed in the **Display Options** sub-section. All of these check boxes are selected by default.

- **Display Measure Description:** When this check box is selected, values that you specify in the **Description** text box in the [Measure sub-section](#) are displayed in the section on the dashboard.

  For example, the following image shows an example **Measures - General** section that has been named *General Measures* when the **Display Measure Description** check box in the **Display Options** sub-section is *selected*.
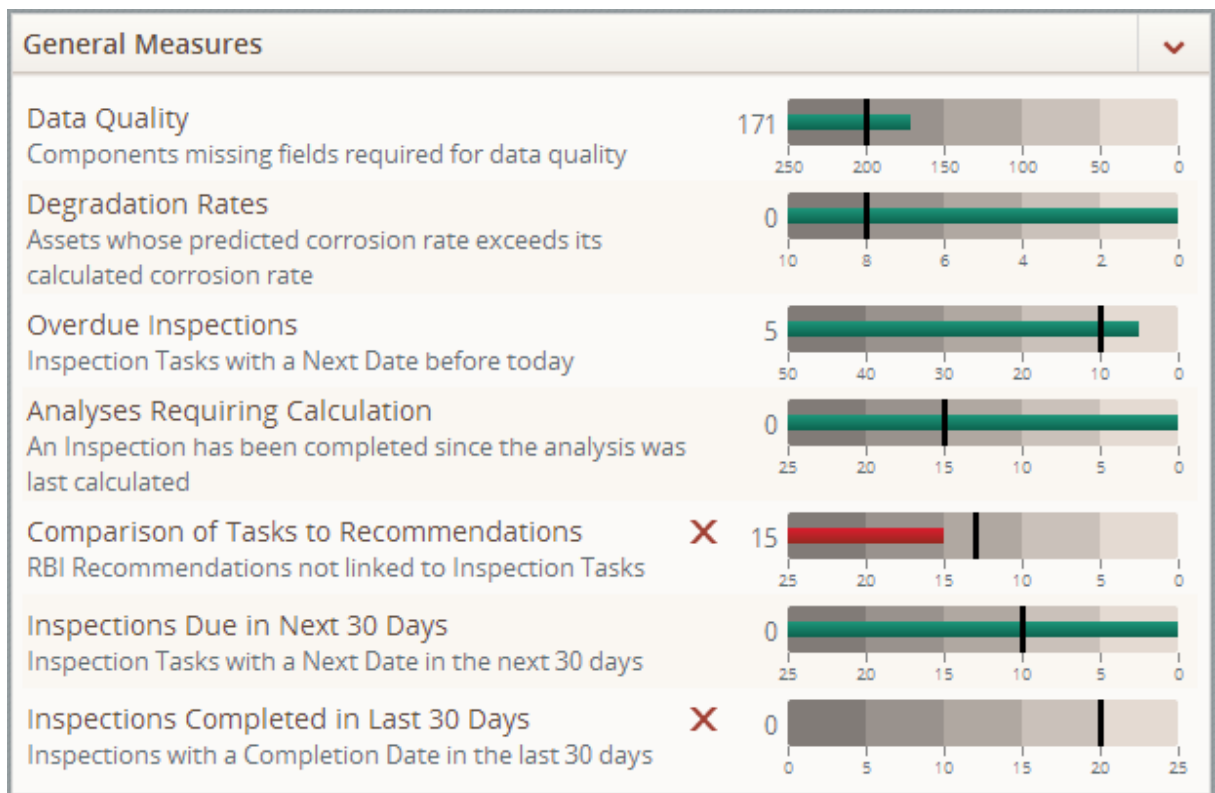
Continuing with the example, the following image shows an example of the **General Measures** section when the **Display Measure Description** check box is *cleared*.
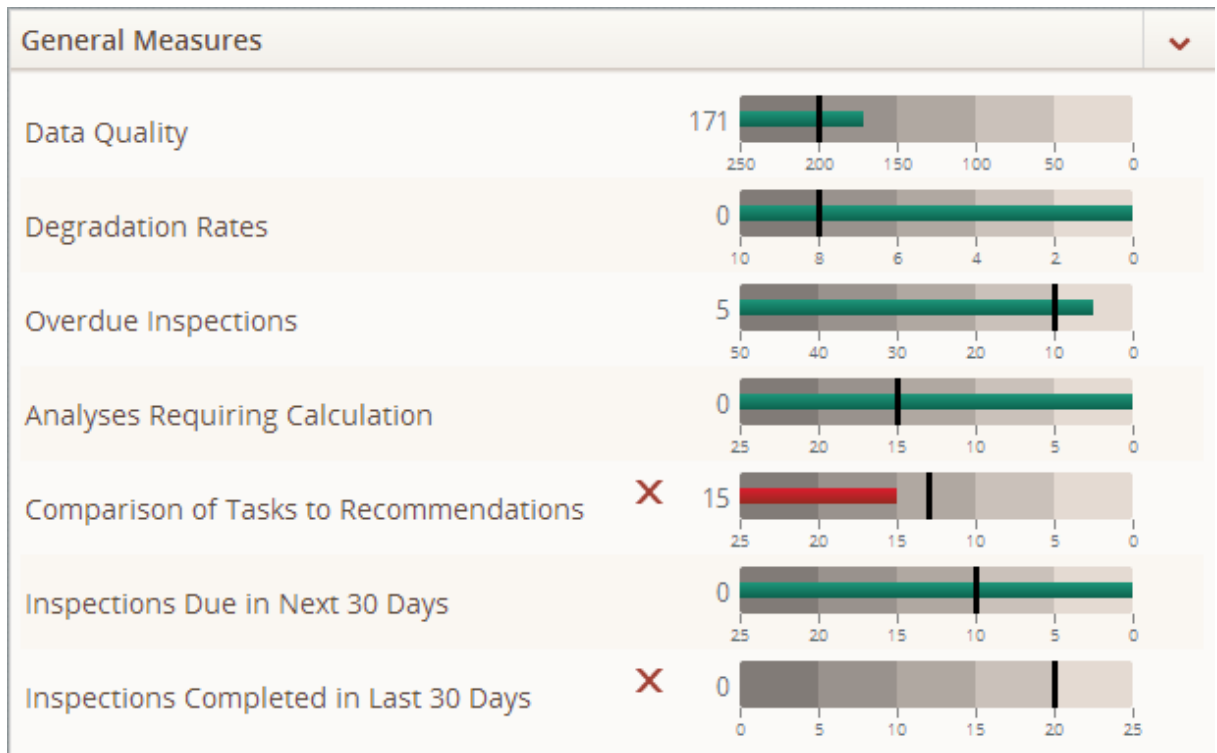
- **Indicate Current Value Meets Target:** When this check box is selected, an ✕ icon is displayed for each measure that fails to meet, exceed, or remain under the [target value](#).

Both of the preceding images in this topic show examples of the **General Measures** section when the **Indicate Current Value Meets Target** check box is *selected*. The following image shows an example of the **General Measures** section when the **Indicate Current Value Meets Target** check box is *cleared*.
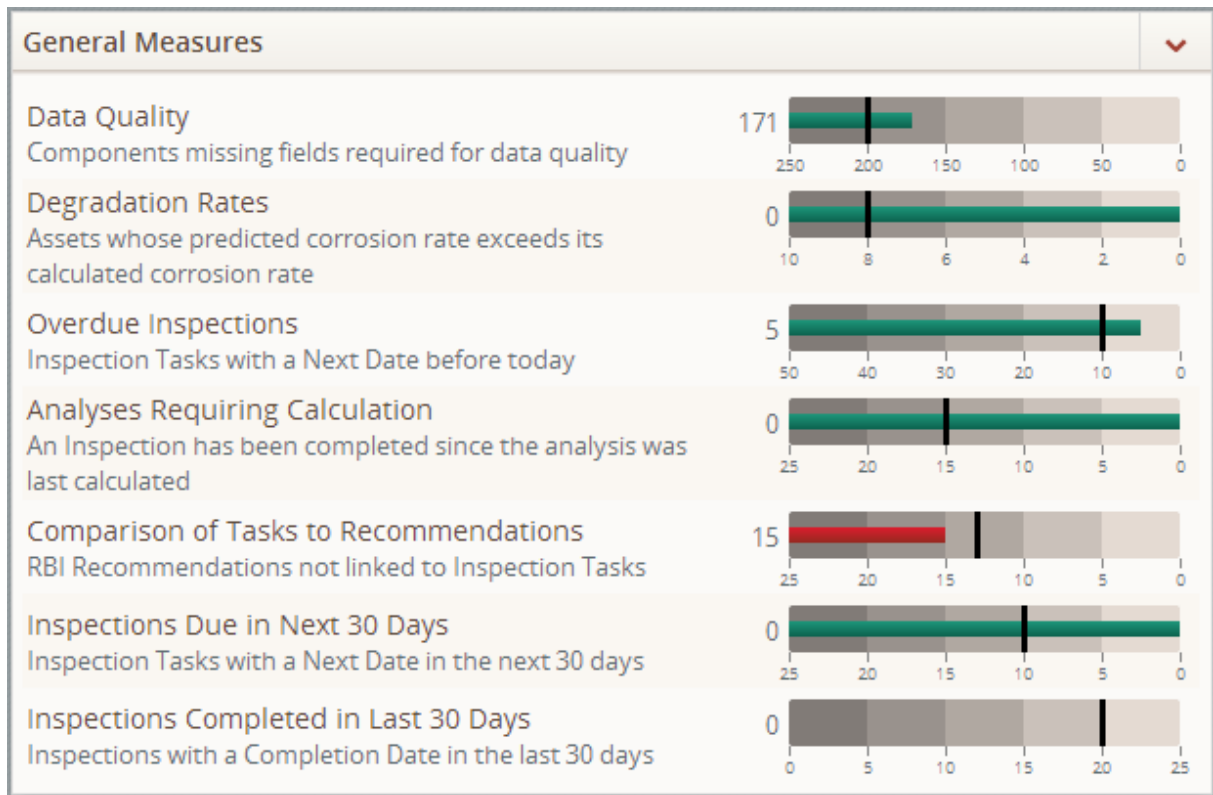
- **Display Current Value As Graph:** When this check box is selected, the Scale and Ranges sub-sections are enabled, and a bar graph is displayed in the dashboard section for each measure that is defined in the edit view. The following color bar graphs are displayed when this check box is selected:
    - **Green:** Indicates that the measure meets or exceeds the target value defined in the Measures sub-section.
    - **Red:** Indicates that the measure does not meet or exceed the target value defined in the Measures sub-section.

All of the preceding images in this topic show examples of the **General Measures** section when the **Display Current Value As Graph** check box is *selected*. The following image shows an example of the **General Measures** section when the **Displays Current Value As Graph** check box is *cleared*.

General Measures		⌄
Data Quality Components missing fields required for data quality		171
Degradation Rates Assets whose predicted corrosion rate exceeds its calculated corrosion rate		0
Overdue Inspections Inspection Tasks with a Next Date before today		5
Analyses Requiring Calculation An Inspection has been completed since the analysis was last calculated		0
Comparison of Tasks to Recommendations RBI Recommendations not linked to Inspection Tasks	✕	15
Inspections Due in Next 30 Days Inspection Tasks with a Next Date in the next 30 days		0
Inspections Completed in Last 30 Days Inspections with a Completion Date in the last 30 days	✕	0

- **Display Current Value:** When this check box is selected, the value that cor-responds with each measure is displayed in one of the following locations in the dashboard section, depending upon your selection in the **Display Current Value As Graph** check box. When the **Display Current Value As Graph** check box is:
  - **Selected:** This value is displayed to the far right of each measure in the sec-tion displayed on the dashboard.
  - **Cleared:** This value is displayed to the left of the bar graph for each measure in the section displayed on the dashboard.

All of the preceding images show examples of the **General Measures** section when the **Display Current Value** check box is *selected*. The following image shows an example of the **General Measures** section when the **Display Current Value** check box is *cleared*.

General Measures

Data Quality
Components missing fields required for data quality

250   200   150   100   50   0

Degradation Rates
Assets whose predicted corrosion rate exceeds its calculated corrosion rate

10   8   6   4   2   0

Overdue Inspections
Inspection Tasks with a Next Date before today

50   40   30   20   10   0

Analyses Requiring Calculation
An Inspection has been completed since the analysis was last calculated

25   20   15   10   5   0

Comparison of Tasks to Recommendations
RBI Recommendations not linked to Inspection Tasks

25   20   15   10   5   0

Inspections Due in Next 30 Days
Inspection Tasks with a Next Date in the next 30 days

25   20   15   10   5   0

Inspections Completed in Last 30 Days
Inspections with a Completion Date in the last 30 days

0   5   10   15   20   25

Edit Measures Sub-Section

The **Edit Measures** sub-section displays various options that you can use to define and modify the content that is displayed in a **Measures - General** section on the dashboard.

The following image shows an example of the **Edit Measures** sub-section.
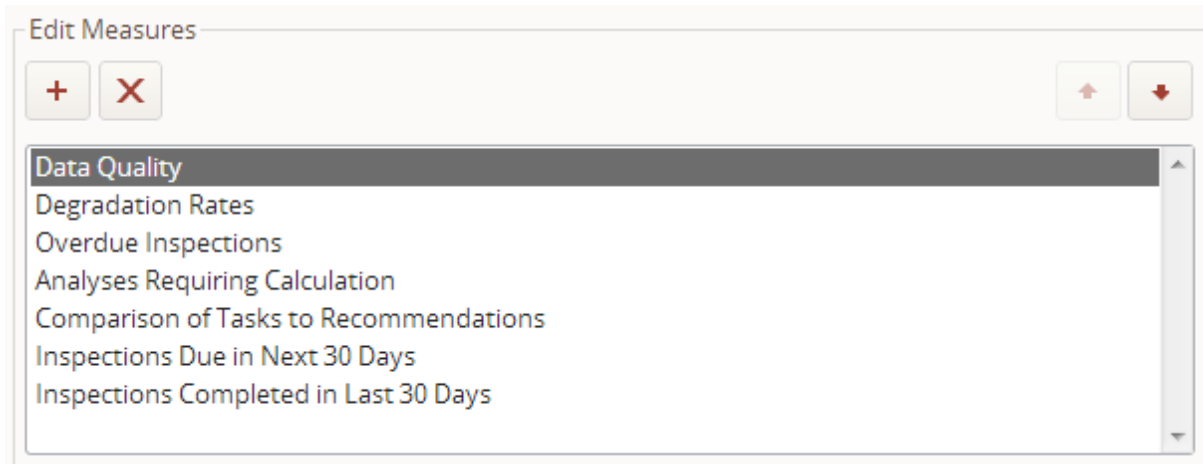
The **Edit Measures** sub-section contains the following items:

- **List of measures:** This unlabeled list displays each measure that you have defined in the **Measure sub-section**. One measure is displayed in the **Measures - General** section on the dashboard for each measure in this list. This list is displayed only when at least one measure has been defined. Throughout this documentation, we refer to this list as the *Measures list*.

- ![+] **button:** Expands the **Measure sub-section**, and adds the value *Measure Name* is added to the **Measures** list.

- ![X] **button:** After asking for confirmation, deletes the measure selected in the **Measures** list.

- ![up] **button:** Moves the selected value down in the **Measures** list. This button is enabled only when more than one value is displayed in the list and when the selected value is not the last value displayed in the list.

- ![down] **button:** Moves the selected value up in the **Measures** list. This button is enabled only when more than one value is displayed in the list and when the selected value is not the first value in the list.

Measure Sub-Section

The **Measure** sub-section is a collapsible sub-section that is used to define attributes for each measure that you added in the **Edit Measures** sub-section.

The following image shows an example of the **Measure** sub-section.

When you access the edit view of the **Measures - General** section for the first time, the **Measure** menu is not displayed. The **Measure** sub-section is:

- Displayed and expanded when you click the ➕ button in the **Edit Measures** sub-section.

- Displayed and expanded any time that you access the edit view of a **Measures - General** section for which one or more measure has been defined in the **Edit Measures** sub-section.

- Collapsed but remains displayed when you click any of the following sub-sections that are displayed in the edit view of the **General - Measures** section:

  - Data Source
  - Scale
  - Ranges

The **Measure** menu contains the following items:

- **Name text box:** The label of the measure that is displayed in the **Measures** list in the **Edit Measures** sub-section. This value corresponds with the measure displayed in the section on the dashboard. The image that follows this list shows an example of a measure that is displayed in a **Measures - General** section on a dashboard that corresponds with the value *Data Quality* specified in the **Name** text box.

- **Description text box:** A description of the measure that is displayed below the measure label in the **Measures - General** section on the dashboard. This value is displayed only when the **Display Measure Description** check box in the **Display**

Options sub-section is selected. The image that follows this list shows an example of a measure that is displayed in a **Measures - General** section on a dashboard that corresponds with the value *Components missing fields required for data quality* specified in the **Description** text box.

- **Target text box and list:** Used to specify the target value for the measure and whether or not a value that is greater than or less than the target value is desirable for the measure.

Target Text Box and List

The **Target** text box and list in the [Measure sub-section](#) are used to specify the target value for a given measure and whether or not a value that is greater than or less than the target value is desirable for the measure.

For example, if a given measure represents the number of overdue inspections in a plant, and your organization's maintenance policy dictates that a plant can have no more than *five* overdue inspections at one time, you would:

- Type the value *5* in the **Target** text box.
- Select the value *Less than target value is good* in the **Target** list.

You *cannot* type a negative target value in the **Target** text box. When the **Display Current Value As Graph** check box is selected in the [Display Options sub-section](#), the target value is displayed as a black line in the bar graph that is displayed for the measure. A red outline has been added to the following image to highlight the black line that is displayed for the target value *10*.
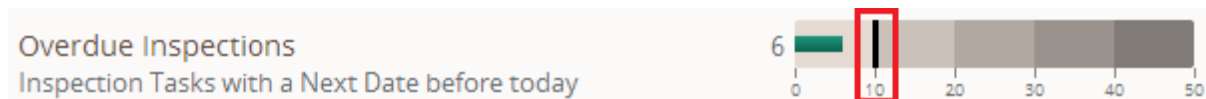
Your selection in the **Target** list will be used to determine the color that is used in the line graph that is displayed for the measure. The list contains the following values:

- **Greater than target value is good:** Indicates that a value for the measure that is *greater* than the target value is desirable. This value is selected by default.

  For example, the following image shows an example of a measure for which the value *Greater than target value is good* has been selected, and the value for the measure, *48*, is greater than the target, *25*. As a result, the line displayed in the graph is green to indicate that the value meets the target (i.e., *48* is greater than *25*).
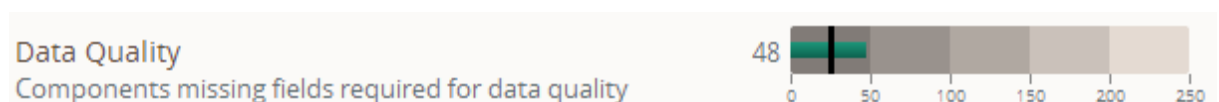
  Alternatively, the following image shows an example of a measure for which the value *Greater than target value is good* has been selected. Note that the value for the measure, *48*, is *not* greater than the target, *200*. As a result, the line displayed in the graph is red, and the ✗ icon appears to the left of the graph to indicate that the value does *not* meet the target (i.e., *48* is not greater than *200*).
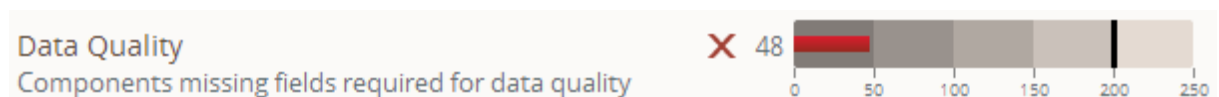
- **Less than target value is good:** Indicates that a value for the measure that is *less* than the target value is desirable.

For example, the following image shows an example of a measure for which the value *Less than target value is good* has been selected. Note that the value for the measure, *6*, is less than the target, *10*. As a result, the line displayed in the graph is green to indicate that the value meets the target (i.e., *6* is less than *10*).

Overdue Inspections
Inspection Tasks with a Next Date before today

6

0   10   20   30   40   50
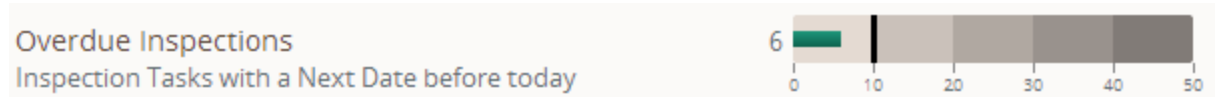
Alternatively, the following image shows an example of a measure for which for which the value *Less than target value is good* has been selected. Note that the value for the measure, 48, is *not* less than the target, *25*. As a result, the line displayed in the graph is red, and the ✕ icon appears to the left of the graph to indicate that the value does *not* meet the target (i.e., *48* is not less than *25*).
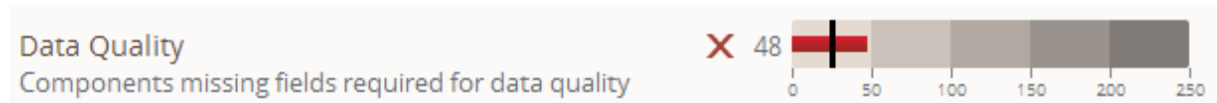
Data Quality
Components missing fields required for data quality

✕ 48

0   50   100   150   200   250

Data Source Sub-Section

The **Data Source** sub-section is a collapsible sub-section that provides options for spe-cifying the source query for each measure defined in the **Edit Measures** sub-section.

The following image shows an example of the **Data Source** sub-section.

To expand the **Data Source** sub-section, you must select a value in the **Measures** list in the **Edit Measures** sub-section. When the **Data Source** sub-section is expanded, if you select a different measure in the **Measures** list, the **Data Source** sub-section collapses, and the **Measure** sub-section is expanded automatically.

The **Data Source** menu contains the following items:

- **Query Path text box:**Defines the Catalog location of the source query whose res-ults will be displayed for the corresponding measure in the section on the dash-board (i.e., the source query). To the right of this text box, the following buttons are displayed:
  - ▪ [...] : Displays the **Choose Query** dialog box where you can select the source query in the Meridium APM Catalog. When you specify the source query using this button, if the query contains prompts, text boxes that correspond with these prompts appear in the **Query Parameters** pane.
  - ▪ [↻] : Refreshes the source query. If you type a value in the **Query Path** text box manually, and the specified query contains prompts, you must click this button in order for the prompt fields to appear in the **Query Parameters** pane.

  If this text box does not contain a valid query path, a validation message appears and a red outline appears around this text box and the **Query Parameters** pane until you specify a valid query path.

- **Query Parameters pane:** If the source query contains prompts, this pane displays text boxes and lists where you should specify values for these prompts. The labels of the text boxes and lists correspond with field captions in records returned by the query, and the values displayed in the lists correspond with the captions of prompt fields. If this pane contains lists and a value is not selected in each list, a validation message appears and a red outline appears around this text box and the **Query Path** text box until you select a value in each list.
- **Measure Value options:** Options for specifying how the selected measure displays the results of the source query. The following items are available:
  - **Query row count option:** When this option is selected, the number of records returned by the source-query is displayed by the selected measure.
  - **Query column option:** When this option is selected, the value in the field specified in the list to the right of this option is displayed by the selected measure. The list to the right of this option contains fields that are returned by the source query.
  - **Show Trend check box:** When this check box is selected, a trend graph is displayed to the right of the bar graph that is displayed for the measure. This check box is enabled only when the **Query column** option is selected.

Trend Graph

You can specify that a trend graph be displayed for a measure by selecting the **Show Trend** check box in the Data Source sub-section. You might want to do this if you want to see a snapshot of the data that has returned in the past by the source query for the measure, in relation to the target value for the measure. When you select the **Show Trend** check box for one or more measures in a **Measures - General** section, the following labels appear above the first measure in the **Measures - General** section:

- **Current:** The bar graphs displayed under this label plot the most recent value returned by the source query for the measure. A red outline has been added to the following image to highlight the **Current** label and the bar graph that is displayed for the measure *Inside Radius*. These graphs are displayed only when the Display Current Value as Graph option is selected in the **Display Options** sub-section.
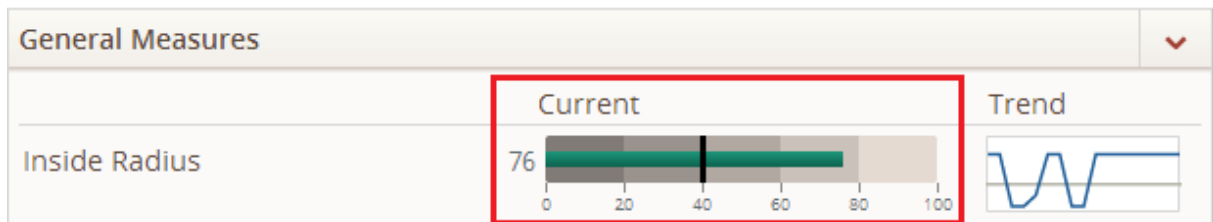
- **Trend:** The line graphs displayed under this label plot the trend of the values returned over time by the source query for the measure. A red outline has been added to the following image to highlight the **Trend** label and the line graph that is displayed for the measure *Inside Radius*.
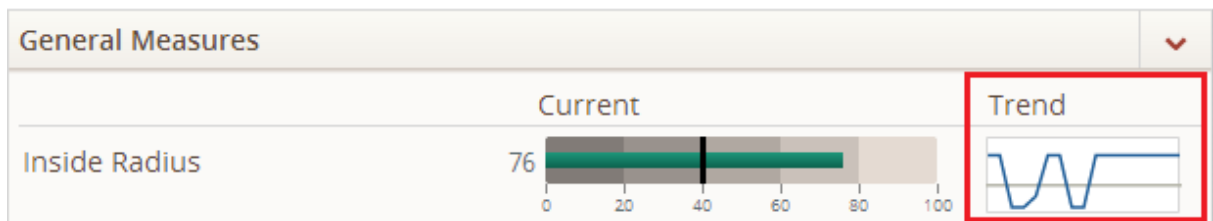
Note that graphs displayed under the **Trend** label do not contain scales. Rather, a horizontal, gray line is displayed in each of these graphs that represents the target value defined for the measure. The vertical placement of the gray line depends upon the scale ranges that are defined for the measure. For example, in the preceding image, the gray line is plotted just below the mid-point of the graph because the scale ranges from *0* to *100*, and the target value is *40* (i.e., the mid-point of this graph would be *50*, and *40* is less than *50*). A second, colored line appears in the graph that shows the values that have been returned by the source query in the past, in relation to the target value (the gray line). The value plotted to the far right on the graph represents the current value returned by the source query for the measure. In the case of the preceding image, the value *76* is plotted to the far right on the graph. This graph is displayed for a measure only if the **Show Trend** check box is selected in the Data Source sub-section for the measure.

Scale Sub-Section

The **Scale** sub-section is a [collapsible sub-section](#) that provides an option for specifying the scale of the [bar graph that can be displayed](#) in the **Measures - General** section on the dashboard. The **Scale** sub-section is enabled only when the **Display Current Value As Graph** check box is selected in the **Display Options** [sub-section](#).

The following image shows what the **Scale** sub-section looks like by default.
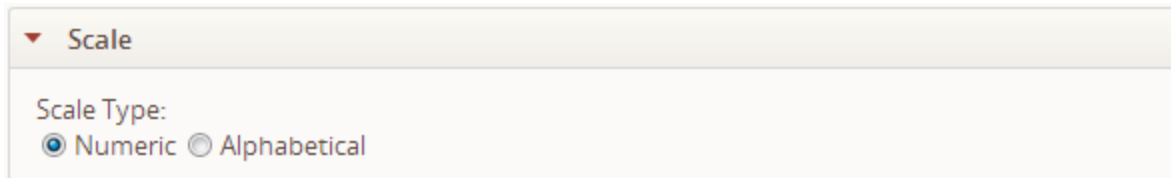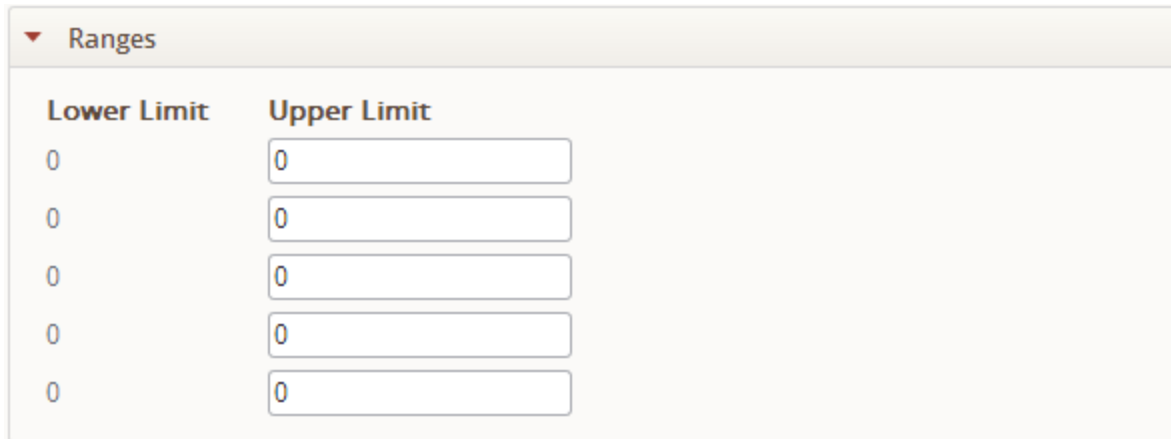
The **Scale** sub-section contains the following **Scale Type** options:

- **Numeric:** You should select this option for measures of numeric values. This option is selected by default.

- **Alphabetical:** You should select this option for measures of text values. Note that when you select this option, the text on the scale will always be displayed in English, regardless of the Culture setting associated with your Security User account.

Ranges Sub-Section

The **Ranges** sub-section is a collapsible sub-section that displays options for configuring the range of values displayed on the scale for bar graphs that can be displayed in the **Measures - General** section on the dashboard. The **Ranges** sub-section is enabled only when the **Display Current Value As Graph** check box is selected in the **Display Options sub-section**.

The following image shows what the expanded **Ranges** sub-section looks like by default.

The **Ranges** sub-section contains the following items:

- **Lower Limit values:** These values are defined according to the values that you specify in the **Upper Limit** text boxes. The first **Lower Limit** value will always be *0* (zero).
- **Upper Limit text boxes:** The values that you type in these text boxes are used to define the **Lower Limit** values for the range of the scale for the bar graph. If you do not specify values for ANY of these text boxes, the graph will not be plotted for the measure. You should specify values in *each* of these text boxes.

For example, when you type a value in the first **Upper Limit** text box, this value is populated as the second **Lower Limit** value. In the following image, the value *5* has been typed in the first **Upper Limit** text box, and the value *5* is populated as the second **Lower Limit** value.

The values that you type in the subsequent **Upper Limits** text boxes are used to populate the subsequent **Lower Limit** values, as shown in the following image.

The following image shows an example of a bar graph that is displayed for a measure according to the range defined above.
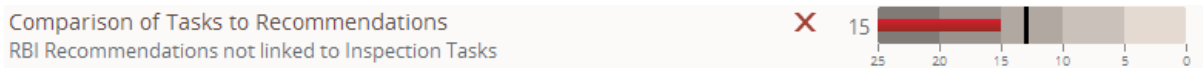
About the Measures - Linked Section

A **Measures - Linked** section on a dashboard displays content that is determined by:

- Various queries and options that are specified in the [edit view of the **Measures - Linked** section](#)

  -and-

- Selections in the following dashboard sections, which can [determine the content that is displayed](#) in a **Measures - Linked** dashboard section:
    - [List - Single Column](#)
    - [List - Summary](#)

The content displayed in a **Measures - Linked** section represents *measures* for assets that you analyze in Meridium APM. Throughout this section of the documentation, the term *measure* is used to refer to a classification of information that you want to analyze about an asset or a group of assets.

The following image shows an example of a **Measures - Linked** section that has been named *Equipment Measures*.
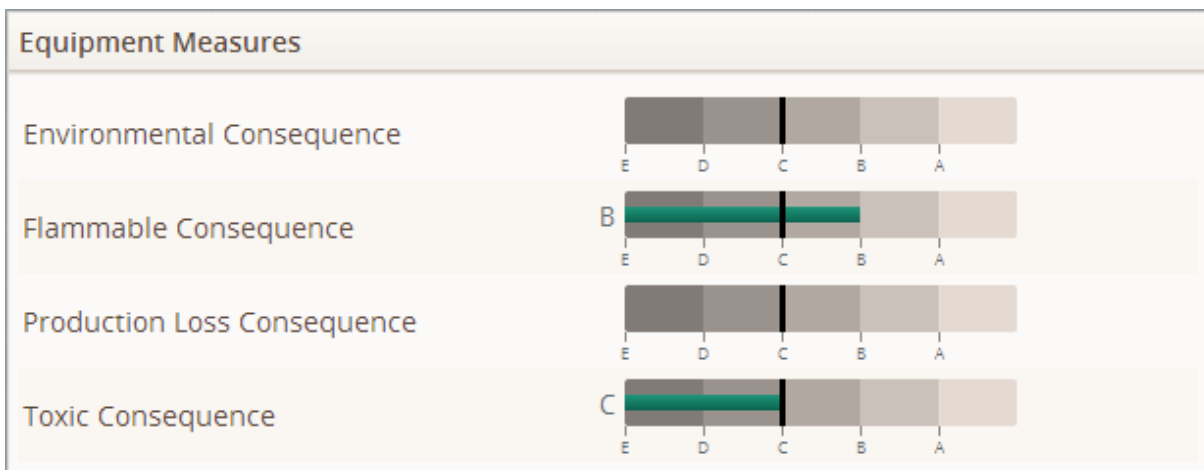
About the Edit View of the Measures - Linked Section

The edit view of the **Measures - Linked** section contains the [same items as the edit view of the **Measures - General** section](#), with the exception of the **Message Channel** text box. The value that you type in this text box specifies the field in the source query used in one of the following *other* dashboard sections whose value will be passed into the prompt contained in the source query in the **Measures - Linked** section in order [to determine the content that is displayed in this section](#):

- [List - Single Column](#)
- [List - Summary](#)

The value that you specify in this text box should *exactly* match the value in the **Message ID** text box in the edit view of the dashboard section in which your selections determine the content that is displayed in the **Measures - Linked** section.

The following image shows an example of the edit view of a **Measures - Linked** section that has been named *Equipment Measures*.

Equipment Measures   🌐   Done

Message Channel:

EquipmentSelected

Display Options

☑ Display Measure Description
☑ Indicate Current Value Meets Target
☑ Display Current Value As Graph
☑ Display Current Value

Edit Measures

+   ✕   ⬆   ⬇

Environmental Consequence
Flammable Consequence
Production Loss Consequence
Toxic Consequence

▼   **Measure**

Name:

Environmental Consequence 🌐

Description

🌐

Target:

3

▶   **Data Source**

▶   **Scale**

▶   **Ranges**

How Selections in Other Dashboard Sections Determine the Content That is Displayed in the Measures - Linked Section
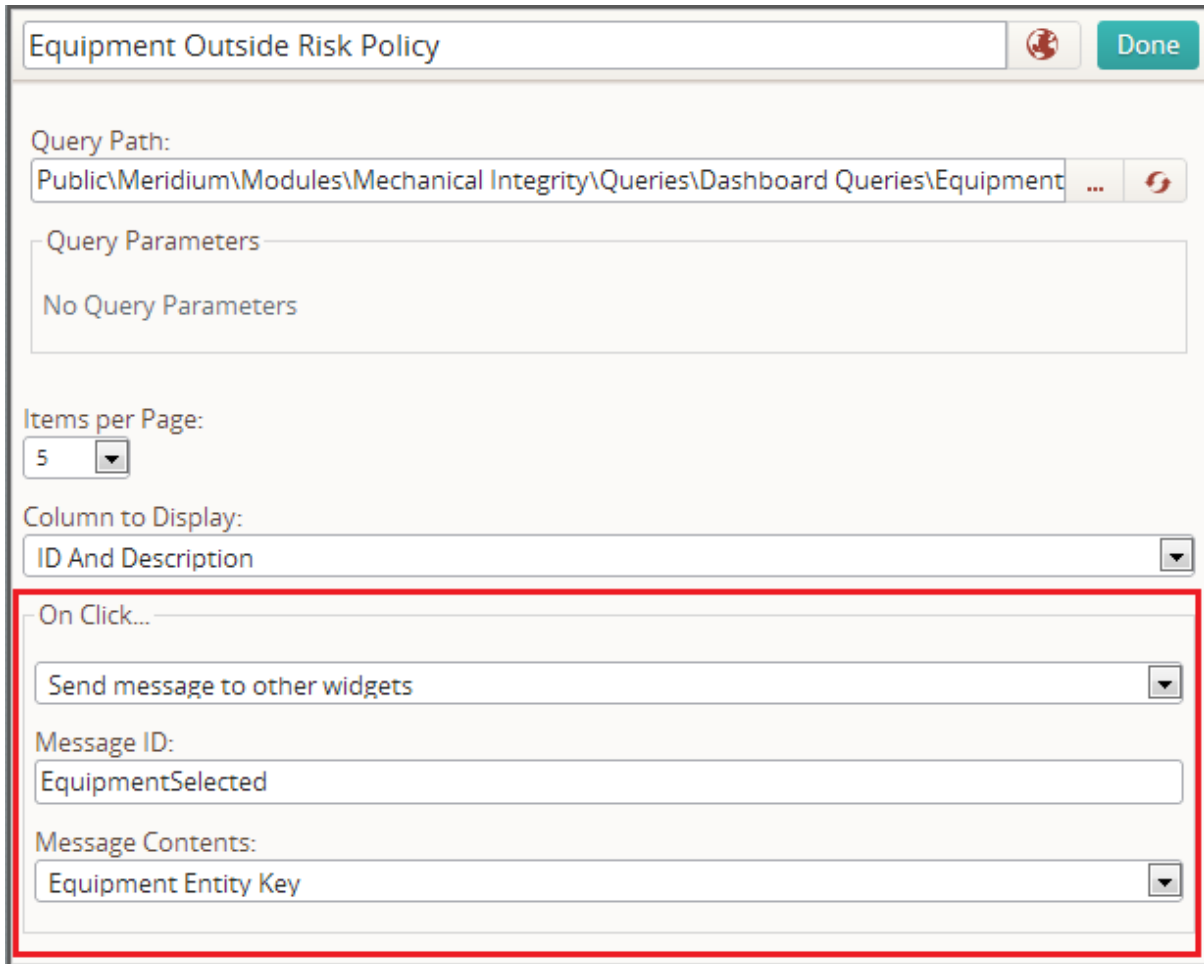
Depending upon your selections in the edit view of the following sections, you can specify that selections in these sections determine the content that is displayed in the **Measures - Linked** section:

- [List - Single Column](#)
- [List - Summary](#)

When you do this, a value returned by the source query used in one of these sections will be passed into a prompt contained in the source query used in the **Measures - Linked** section in order to determine the content that is displayed in the **Measures - Linked**section.

For example, consider a dashboard that contains the **Equipment Outside Risk Policy** section (a configured **List - Single Column** section) that displays a list of Equipment records that are associated with RBI Criticality Analysis records that meet certain criteria. Now consider that the same dashboard contains the **Equipment Measures** section (a configured **Measures - Linked** section) that displays graphs for various measures, which correspond with values in RBI Criticality Analysis records that are associated with Equipment records. You can define configurations in the edit view of these sections such that your selections in the **Equipment Outside Risk Policy** section will determine he content that is displayed in the **Equipment Measures** section:

Specifically, in the edit view of the **Equipment Outside Risk Policy** section, you would specify the values in the fields in the **On Click** pane shown in the following image. A red outline has been added to the image to highlight the **On Click** pane.

Next, in the edit view of the **Equipment Measures** section, you would type the value *EquipmentSelected* in the **Message Channel** text box, as shown in the following image. A red outline has been added to the image to highlight the **Message Channel** text box.

Note that:

- In the edit view of the **Equipment Outside Risk Policy** section:
  - The value *Send messages to other widgets* is selected in the list.
  - The **Message ID** text box contains the value *EquipmentSelected*.
  - The value *Equipment Entity Key* is selected in the **Message Contents** list.
- In the edit view of the **Equipment Measures** section, the **Message Channel** text box contains the value *EquipmentSelected*.
- The value in the **Message Channel** text box in the edit view of the **Equipment Measures** section exactly matches the value in the **Message ID** text box in the edit view of the **Equipment Outside Risk Policy** section.

Based upon this configuration, when an Equipment record is selected in the **Equipment Outside Risk Policy** section, the Entity Key of that Equipment record is passed into the Entity Key prompt in the source query in the **Equipment Measures** section, and as a result, the **Equipment Measures** section displays information specific to that Equipment record.

Continuing with the example, consider the following image that shows an example of the dashboard containing these sections when an Equipment record is not selected in the **Equipment Outside Risk Policy** section.
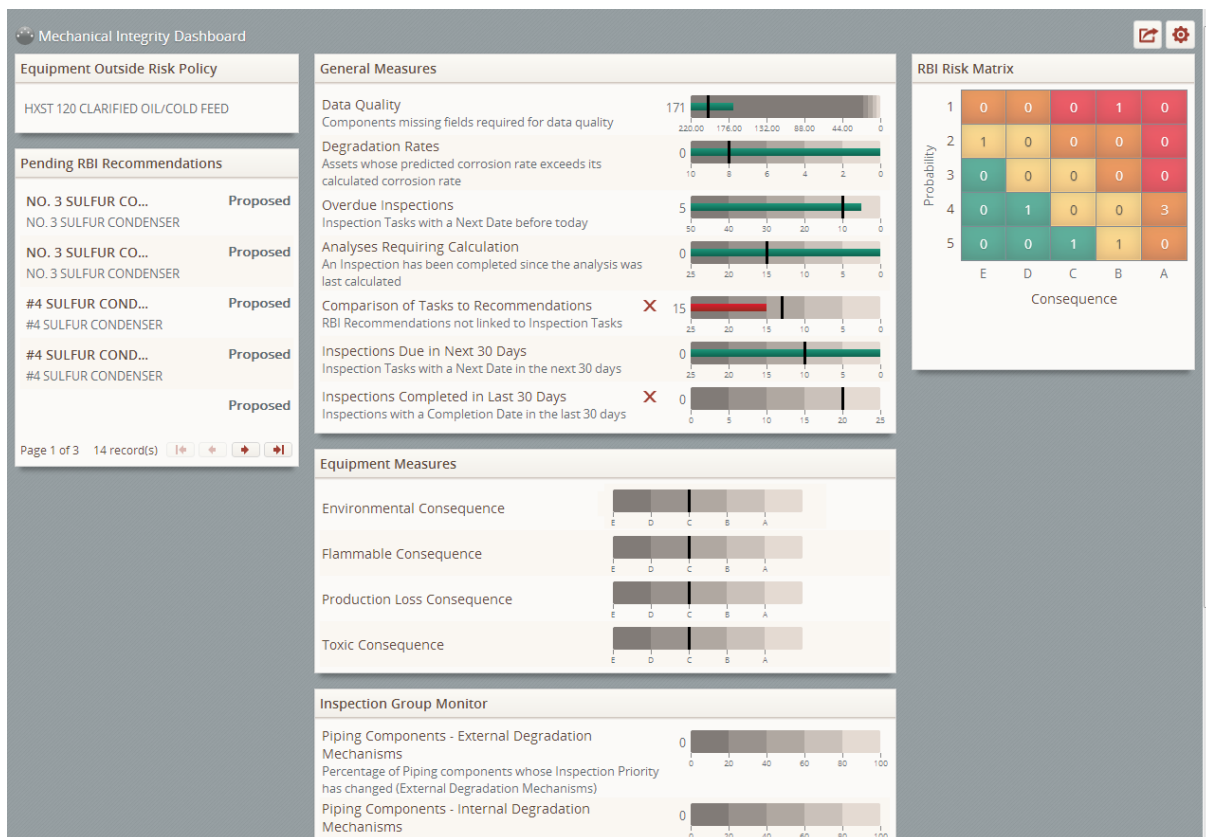
Now consider the following image that shows an example of the dashboard containing these sections when an Equipment record in the **Equipment Outside Risk Policy** section is *selected*.
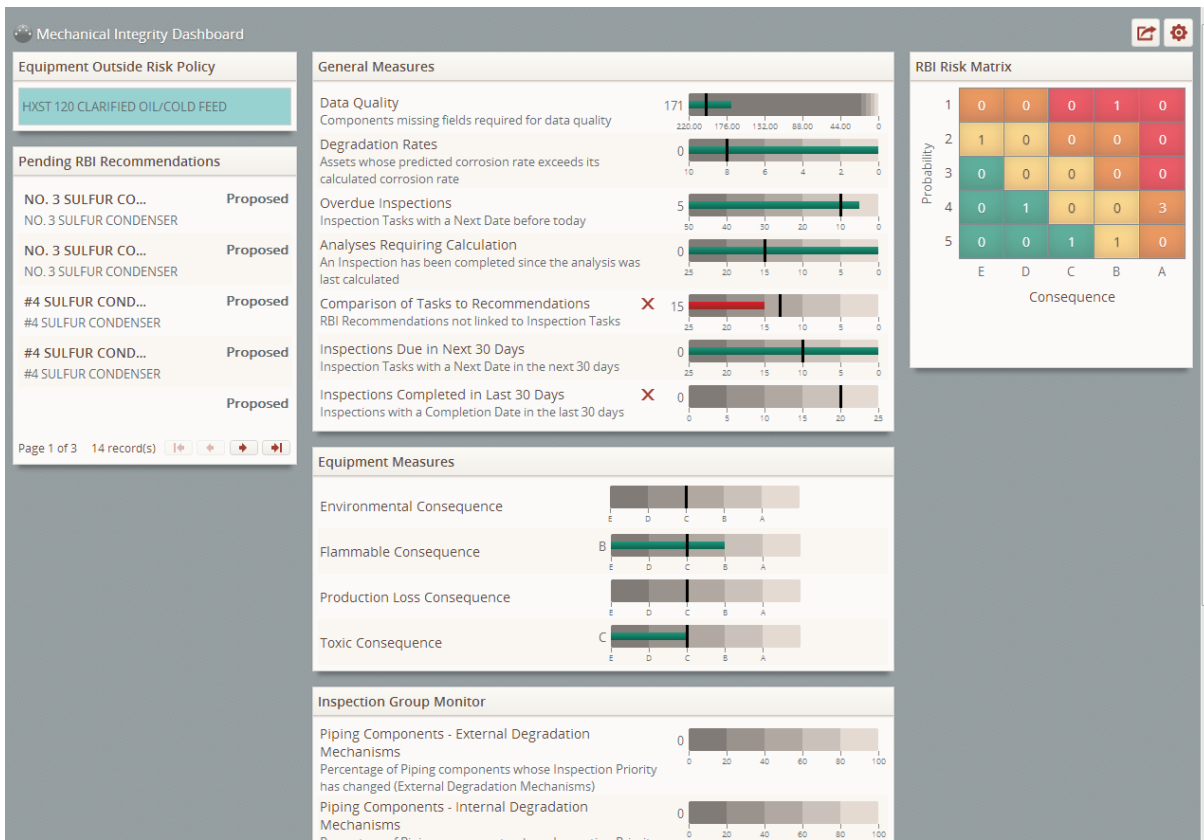
About Collapsible Sub-Sections

The edit view of the following dashboard sections contain collapsible sub-sections that are used to define content that is displayed in that section on the dashboard:

- Chart - Series
- Measures - General
- Measures - Linked

The following image shows an example of what collapsible sub-sections look like in the edit view of a **Measures - General** dashboard section.
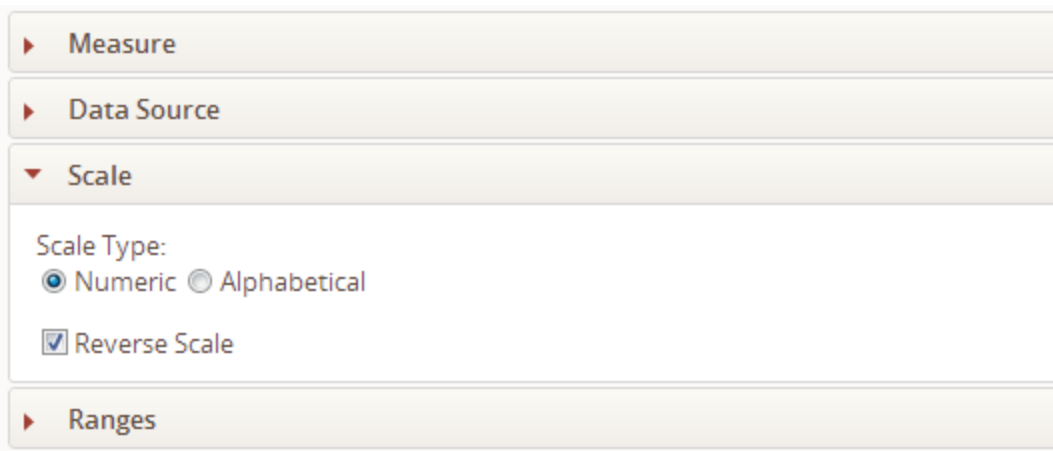
In the above image, the **Scale** sub-section is expanded, and the following sub-sections are collapsed:

- **Measures**
- **Data Source**
- **Ranges**

This section of the documentation provides details on expanding and collapsing collapsible sub-sections.

Expanding and Collapsing Collapsible Sub-Sections

The following instructions assume that you have accessed the edit view of a [dashboard section that contains collapsible sub-sections](#).

To expand a collapsible sub-section:

- Click the header of the collapsible sub-section that you want to expand.

  The sub-section is expanded, and the background color of the sub-section header changes to white.

  For example, the following image shows an example of the **Scale** sub-section expanded.
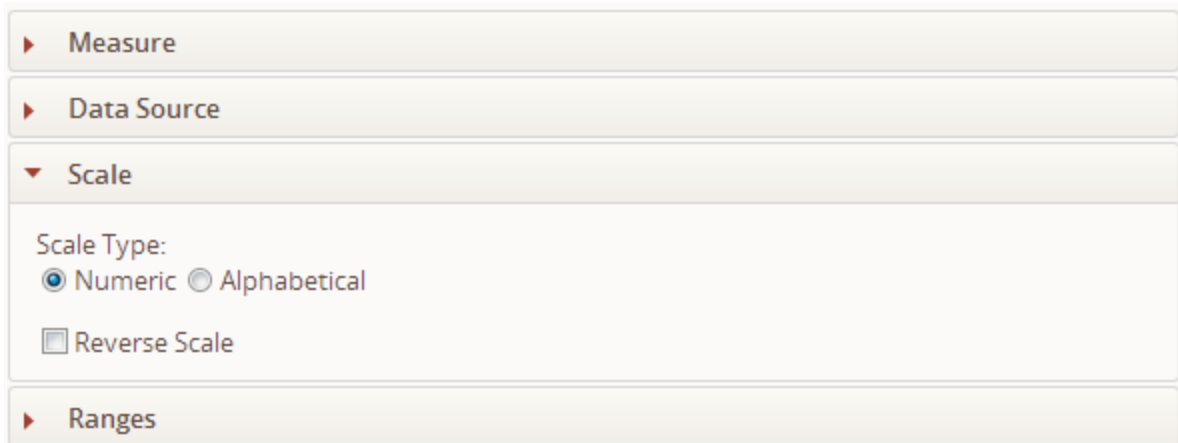
If you clicked the section header of the **Ranges** sub-section, the **Scale** sub-section would be collapsed, and the **Ranges** sub-section would be expanded, as shown in the following image.

Measure		
Data Source		
Scale		

▼ Ranges

Lower Limit	Upper Limit
0	50
50	100
100	150
150	200
200	250

To collapse a collapsible sub-section:
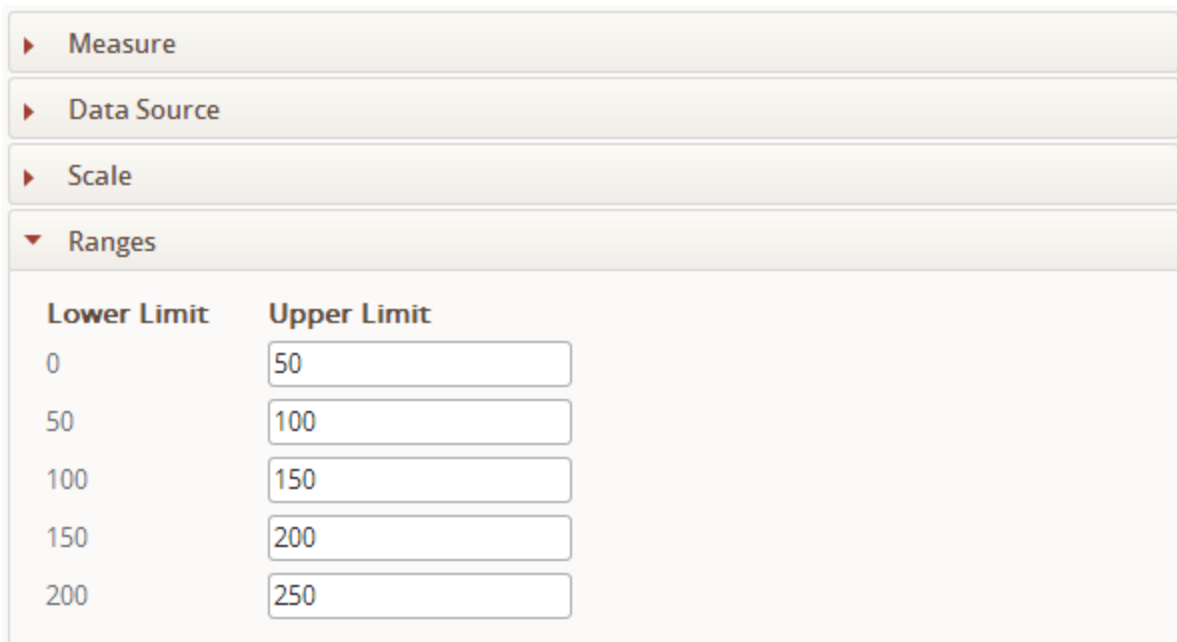
- Click the header of another collapsible sub-section.

  The selected sub-section is expanded, and the other sub-section is collapsed. The background color of the previously expanded sub-section header changes to blue.

  For example, the following image shows an example of the **Measure** sub-section expanded.

▼ Measure

Name:

Data Quality

Description

Components missing fields required for data quality

Target:

200

Data Source		
Scale		
Ranges		

If you clicked the section header of the **Ranges** sub-section, the **Measures** sub-section would be collapsed and the **Ranges** sub-section would be expanded, as shown in the following image.

▸ Measure

▸ Data Source

▸ Scale

▾ Ranges

Lower Limit	Upper Limit
0	50
50	100
100	150
150	200
200	250

About Query Results Displayed in Dashboard Sections

The following dashboard sections display grids or lists that contain the results of a query specified in the edit view of the section:

- List - Query Result
- List - Single Column
- List - Summary

Whether a grid or a list is displayed, and in the case of grids, the number of columns displayed and their labels, will vary depending upon the type of dashboard section. Regardless of the type of section, one row is displayed in a grid and one value or group of values is displayed in a list for each record returned by the query. If a query is configured to contain hyperlinks, the hyperlinked values will be displayed in the grid or list displayed in the section.

When the results of the query are displayed on *more than one* page in the section, the following items are displayed at below the grid or list:

- **Page count:** The text *Page <current page> of <total pages>* is displayed, where <current page> is the number of the current page of query results and <total pages> is the total number of pages of query results.
- **Record count:** The text *<number> record(s)* is displayed, where <number> is the total number of records returned by the query.
- The following buttons are displayed:
    - |◄ : Displays the first page in the query results. This button is enabled only when the current page is any page *other than* the first page in the query results.
    - ◄ : Displays the previous page in the query results. This button is enabled only when the current page is any page *other than* the first page in the query results.
    - ► : Displays the next page in the query results. This button is enabled only when the current page is any page *other than* the last page in the query results.
    - ►| : Displays the last page in the query results. This button is enabled only when the current page is any page *other than* the last page in the query results.

Saving Dashboards

When you work with dashboards on the **Dashboard page**, all modifications that you make are saved automatically by the Meridium APM system. When you work with the edit view of dashboard sections, you should click the **Done** button to close the edit view, even though changes that you make in the edit view are saved automatically.

Deleting Dashboards

You can delete a dashboard in the same way that you can delete other Catalog items. In order to delete a dashboard, you must have [Delete privileges to the Catalog location](#) in which the dashboard is stored.

About Working with Existing Dashboards

If you have the [necessary security permissions](#), you can perform the following actions on existing dashboards:

- [Hide sections](#) that you do not want to see or that are not relevant to your personal work process.
- [Redisplay hidden sections](#).
- [Modify the number of columns displayed on the dashboard or their layout](#).
- [Arrange the dashboard sections](#).
- [Modify the height of dashboard sections](#).
- [Remove your user display preferences](#).
- [Restore dashboard sections to their default size](#).
- [View details about data that is displayed in a dashboard section](#).
- [Filter dashboard content](#).
- [Refresh the dashboard](#).

About User Display Preferences

The term *user display preferences* is used to refer to the way that a dashboard looks when you have done any of the following:

- Hidden one or more dashboard sections.
- Redisplayed one or more hidden dashboard sections.
- Modified the height of one or more dashboard sections.
- Modified the number of columns or column layout of a dashboard.
- Arranged dashboard sections.
- Restored a dashboard section to its default size.

Your user display preferences are retained when another user makes either of the following modifications to the dashboard:

- Adds a new section.
- Modifies the content of an existing section.

When another user *removes* a section from the dashboard for which you have specified display preferences, however, your display preferences will be retained, but the section that was removed will no longer be displayed.

Hiding Sections on a Dashboard

If a dashboard displays sections that are not relevant to you or that you do not want to see, you can hide these sections. Any section that is hidden can also be re-displayed. The following instructions assume that you have accessed the configuration buttons on the dashboard whose sections you want to hide.

To hide a section on a dashboard:

1. At the top of the section that you want to hide, click the ⌄ button.

   A submenu appears. A red outline has been added to the following image to high-light the submenu that is displayed to a user that has additional privileges beyond View privileges to the Catalog folders that store the dashboard and its underlying queries.
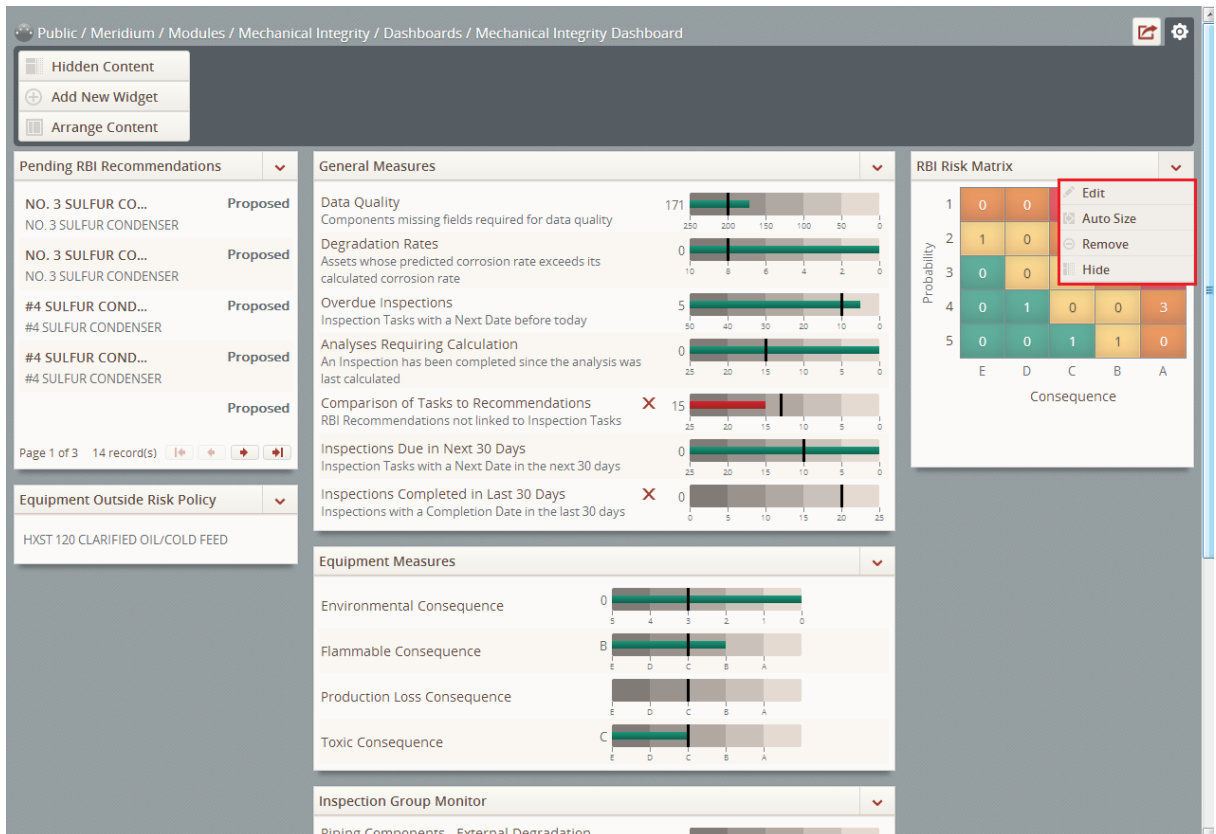
2. Click the **Hide** option.

   A confirmation message appears, asking if you are sure that you want to hide the section.

3. Click the **Hide** button.

   The dashboard section is hidden.

The name of the section is added to the **Hidden Content** list.

Redisplaying Hidden Sections on a Dashboard

If you want to view a dashboard section that has been hidden, either by yourself or another user, you can redisplay the section. A maximum of 12 sections can be displayed on the dashboard at one time. For optimal performance and usability, however, we recommend that you display *no more than* eight sections on a dashboard at one time. The following instructions assume that you have accessed the configuration buttons on the dashboard whose hidden sections you want to redisplay.

To re-display a hidden section on a dashboard:

1. Click the **Hidden Content** button.

   The list of hidden sections and the buttons used to redisplay these sections on a dashboard appear. The following image shows an example of a dashboard whose **RBIRisk Matrix** section has been hidden.
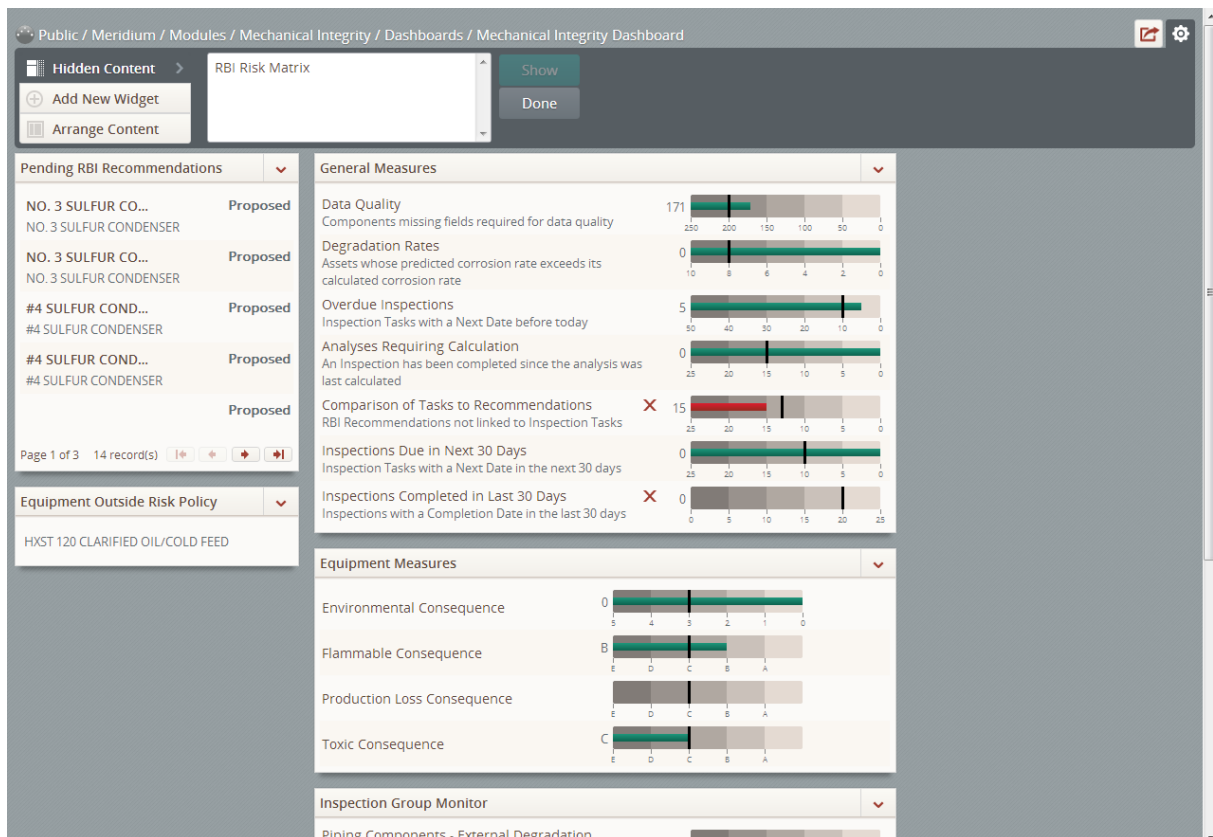
2. In thelist of hidden sections, select the section that you want to redisplay.

3. Click the **Show** button.

   The section appears in the first column on the left of the dashboard, and the section name is removed from the list of hidden sections.

Note: If the dashboard for which you redisplayed a hidden section already displays between eight and 11 sections, a warning message appears to the right of the **Show** button, indicating that you should not display more than eight sections on a dashboard.

4. Click the **Done** button.

The list of hidden sections and the buttons used to redisplay these sections disappear.

Modifying the Number of Columns or Column Layout

You can modify the number of columns or the column layout that is used to display sections on the Dashboard page. The following instructions assume that you have accessed the configuration buttons on the dashboard whose number of columns or column layout you want to modify.

To modify the number of columns or column layout:

1. Click the **Arrange Content** button.

   - The column options and the **Done** button appear.
   - The column option that corresponds with the current layout of the dashboard is selected.
   - Dotted lines appear that denote the current layout of the dashboard.

   The following image shows an example of a dashboard with a three column option selected.
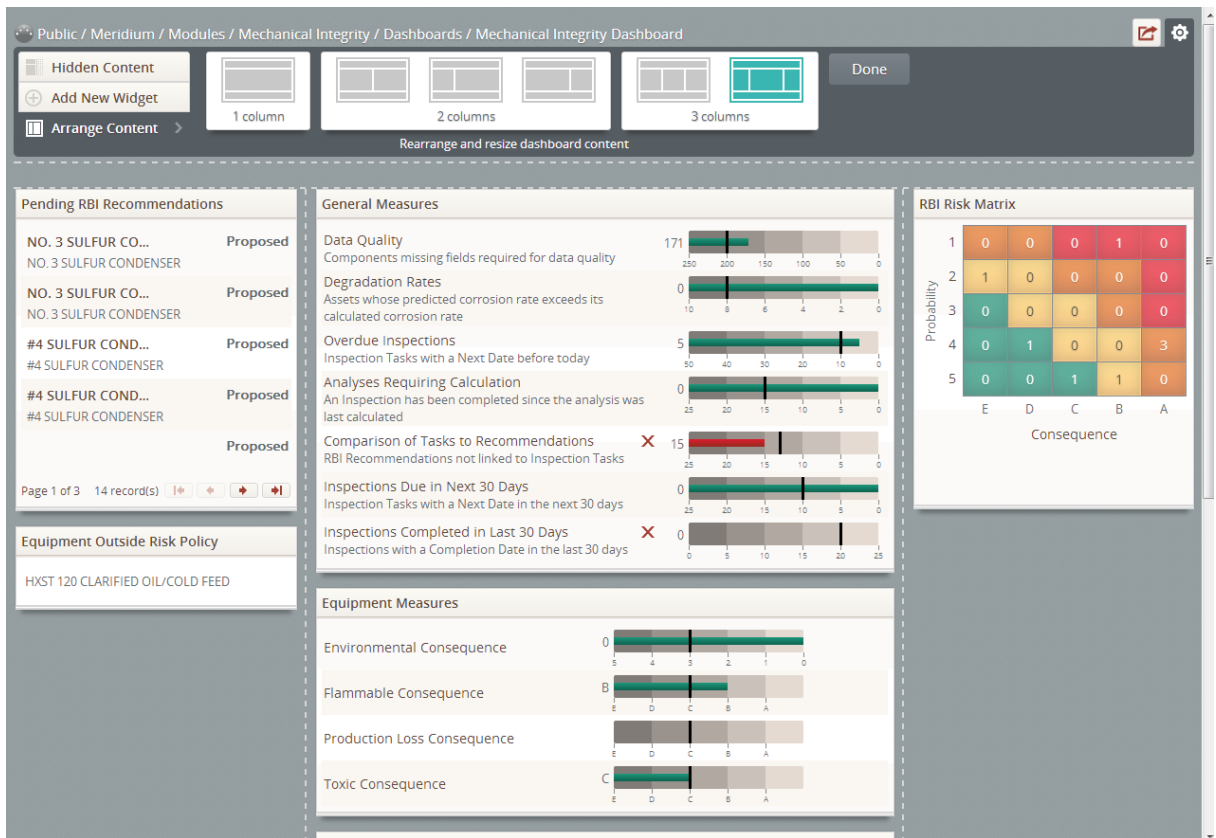
2. Select the column option that you want to be applied to the dashboard layout.

Your selection is applied automatically to the dashboard, and the dashboard sections are re-sized automatically according to your selection. In the following image, a different three column option has been selected.

3. Click the **Done** button.

   The column options and the dotted lines disappear.

Arranging Dashboard Sections

You can arrange dashboard sections within the columns and rows on a dashboard. The following instructions assume that you have accessed the dashboard configuration buttons on the dashboard for which you want to arrange dashboard sections.

To arrange dashboard sections within columns and rows:

1. Click the **Arrange Content** button.

   - The column options and the **Done** button appear.
   - The column option that corresponds with the current layout of the dashboard is selected.
   - Dotted lines appear that denote the current layout of the dashboard.

   The following image shows an example of a dashboard with a three column option selected.
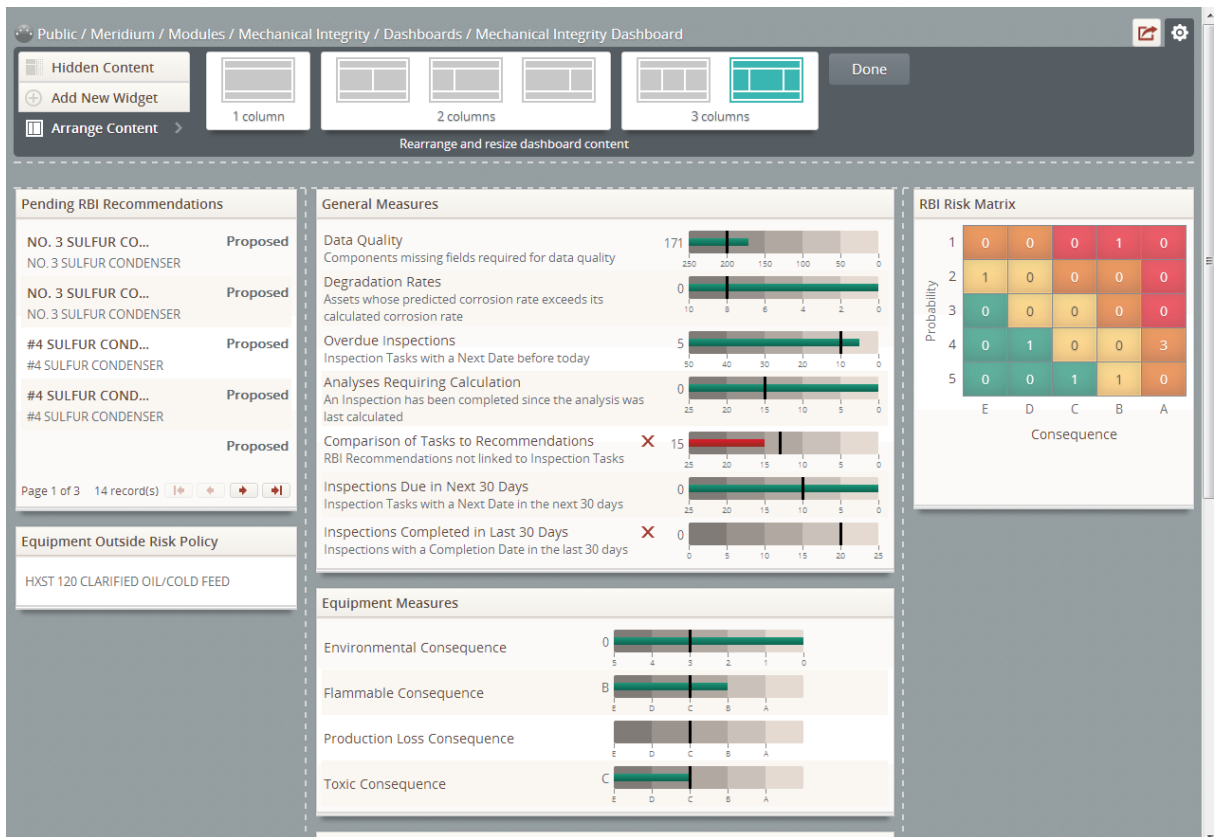
2. Pause on the label of the section that you want to move to a different column or row.

   The pointer appears as ⊕.

3. Click the section that you want to move, and point to the new location where you want the section to appear.

    A shaded area appears on the screen where the section will be positioned after you release the mouse button.

4. Release the mouse button.

    The section appears in the new location, and the pointer no longer appears as ✥. The following image shows how the **Equipment Outside Risk Policy** section has been moved from the first column on the left side of the dashboard to the first column on the right side of the dashboard.

5. Click the **Done** button.

    The column options and dotted lines disappear.

Modifying the Height of Dashboard Sections

The following instructions assume that you have accessed the dashboard configuration buttons on the dashboard whose section height you want to modify.

To modify the height of a dashboard section:

1. Click the **Arrange Content** button.

   - The column options and the **Done** button appear.

   - The column option that corresponds with the current layout of the dashboard is selected.

   - Dotted lines appear that denote the current layout of the dashboard.

     The following image shows an example of a dashboard with a three column option selected.
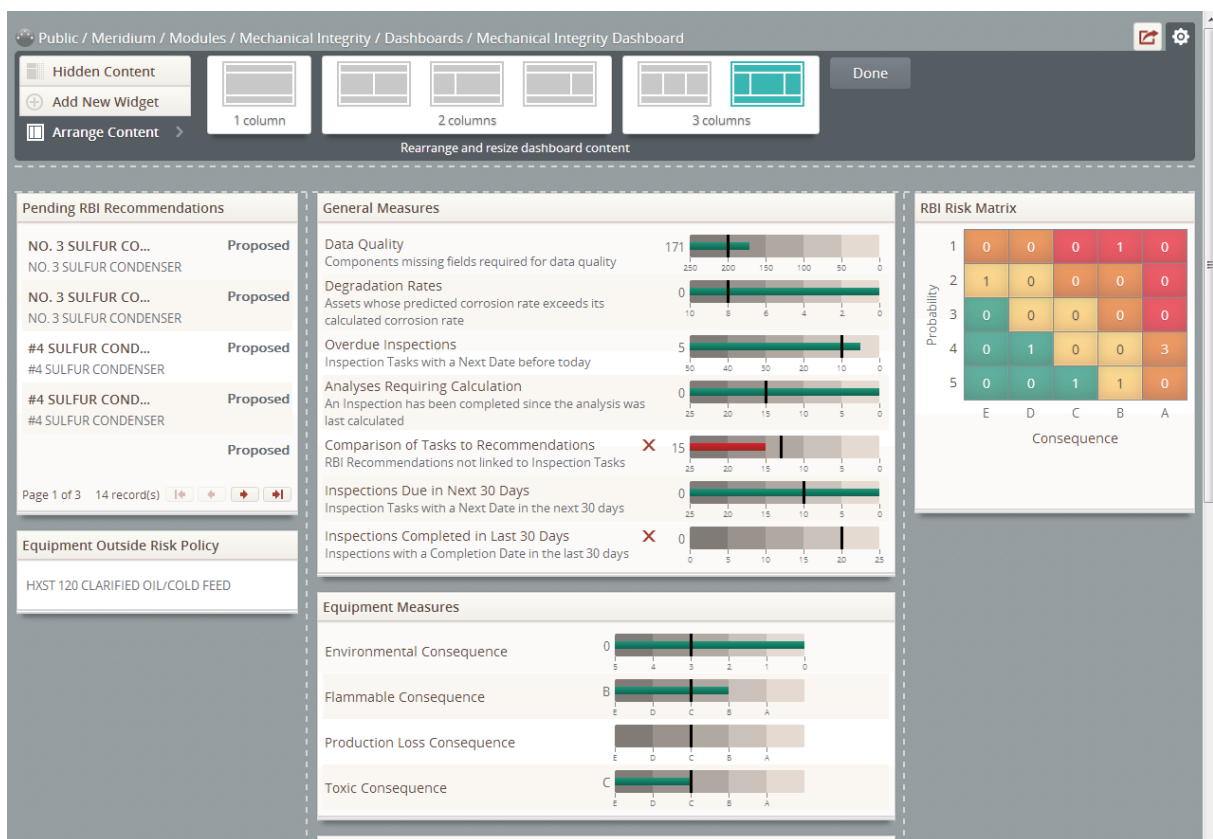
2. Pause on the bottom border of the dashboard section whose height you want to modify.

   The pointer appears as ⇕.

3. Click the bottom border of the dashboard section, and drag the border until you reach the height that you want.

4. Release the mouse button.

The section is displayed according to height that you specified, and the pointer no longer appears as ↕.

The following image shows an example of **Pending Recommendations** section, which has been decreased in height.

5. Click the **Done** button.

The column options and dotted lines disappear.

Restoring Dashboard Sections to Their Default Size

If you have modified the height of a dashboard section, you can restore it to its default size. The following instructions assume that you have accessed the dashboard configuration buttons on the dashboard containing a section whose size you want to restore.

To restore a dashboard section to its default size:

1. At the top of the section that you want to restore to its default size, click the ⌄ button.

    A submenu appears. A red outline has been added to the following image to highlight the submenu that is displayed to a user that has additional privileges beyond View privileges to the Catalog folders that store the dashboard and its underlying queries.
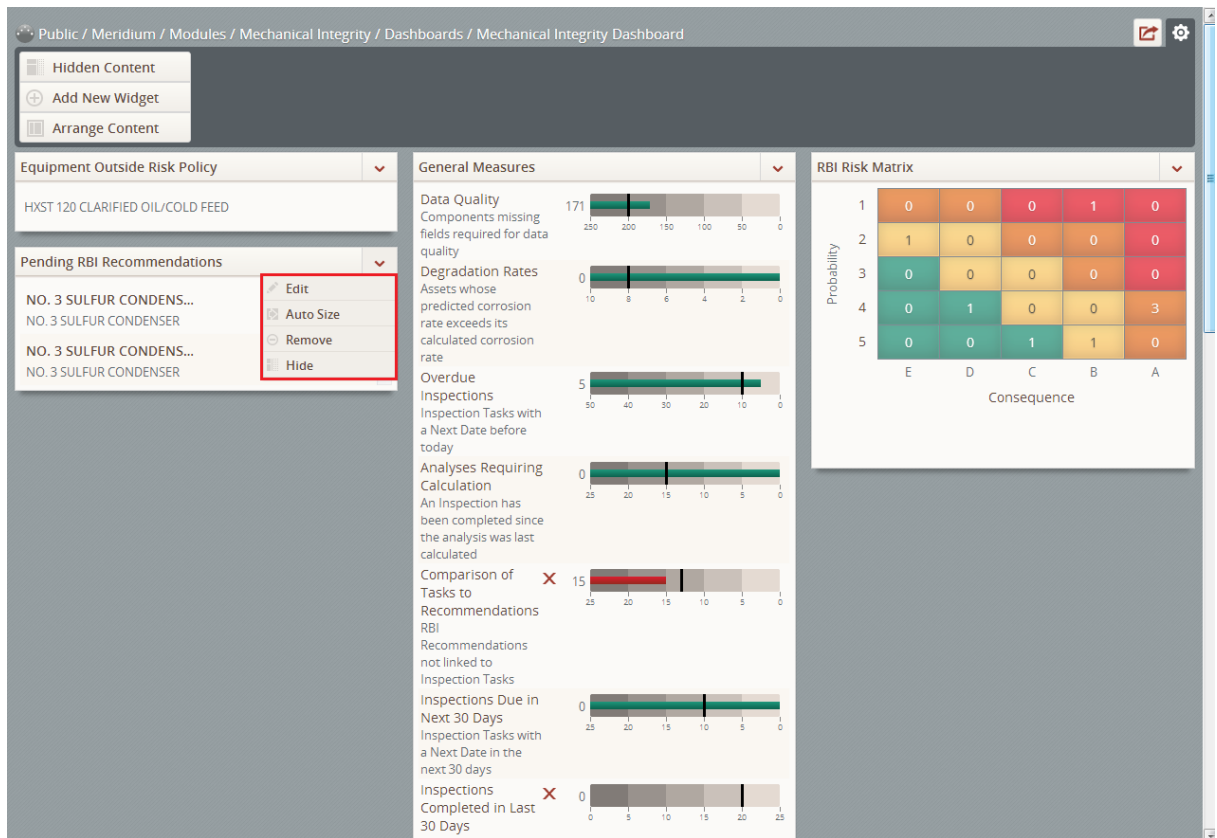
2. Click the **Auto Size** option.

3. The size of the section is restored automatically.

Clearing User Display Preferences

You can remove all of your user display preferences at one time in order to reset the dashboard according to the specifications of the user who last saved the dashboard. The following instructions assume that you have accessed the dashboard configuration buttons on the dashboard for which you want to reverse changes that you have made to the display.

To remove your user display preferences:

1. Click the **Revert Changes** button.

   A confirmation message appears.

2. Click the **Revert** button.

   All of your user display preferences are removed, and the dashboard is displayed according to the specifications of the user who last saved it.

Viewing Details for Dashboard Section Data

When you click values that are displayed in some dashboard sections, a dialog box appears, displaying a grid that contains details about that data. The label and content of this dialog box will vary, depending upon the dashboard section in which you clicked the values and the query that is configured in the edit view of the section. In the dialog box, each column in the grid will correspond with a field returned by the query in the edit view of the section. Throughout the documentation, we refer to these dialog boxes as *detail dialog boxes.*

Some dashboard sections display detail dialog boxes by default when you click data in the section, and for some dashboard sections, you must specify manually that a detail dialog box be displayed. The following table lists the dashboard sections and information about the detail dialog that appears for each section.

Dashboard section	Displays a detail dialog box (Yes/No)	Notes
Chart-Series	Yes	A detail dialog box is displayed when you click a datapoint in the chart.
Content Filter	No	N/A
Hyperlinks	No	N/A
List - Query Result	No	N/A
List - Single Column	Yes	When the value *View Detail* is selected in the **On Click...** list in the edit view of the section, a detail dialog box is displayed when you click a value that is displayed in the section.
List - Summary	Yes	When the value *View Detail* is selected in the **On Click...** list in the edit view of the section, a detail dialog box is displayed when you click a value that is displayed in the section.
Measures - General	Yes	A detail dialog box is displayed when you click a measure that is displayed in the section.

Dashboard section	Displays a detail dialog box (Yes/No)	Notes
Measures - Linked	Yes	A detail dialog box is displayed when you click a measure that is displayed in the section.
RBI - Risk Matrix	Yes	A detail dialog box is displayed when you click a value in the Risk Matrix. The **RBI - Risk Matrix** section is displayed only on the baseline Mechanical Integrity dashboard and is for use only on dashboards that are used by Mechanical Integrity users.

Filtering Dashboard Content

The following instructions assume that you are viewing a dashboard whose content you want to filter, and that the dashboard contains sections whose content you can filter.

To filter content that is displayed on a dashboard:

1. In the **Content Filter** section, in the list or text box that corresponds with the field on which you want to filter content in other dashboard sections, type or select the value on which you want to filter.

2. Click the **Apply** button.

   The content of dashboard sections is filtered, depending upon the values that you specified.

   For example, the following image shows an example of a dashboard where the following values have been specified for the following items:

- **Route ID list:** *ALL*
- **Asset ID list:** *ALL*
- **Compliance Start Date text box:** *12/03/2012 14:38:40*
- **Compliance End Date text box:** *05/08/2013 14:38:40*

Continuing with the example, the following image shows an example of a dashboard when the values selected in the *Route ID* and *Asset ID* lists were changed from *ALL* to *10 R* and *PMP-3830* respectively.
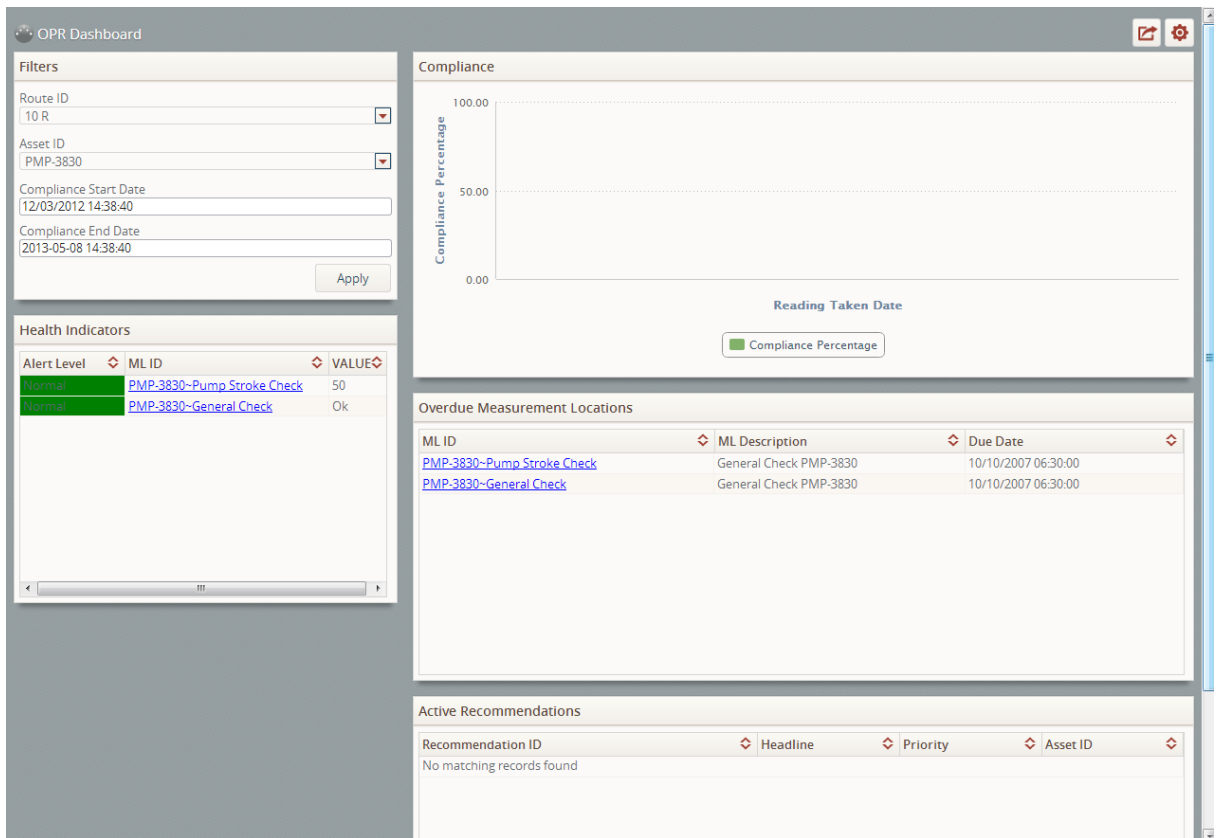
Viewing Details for Datapoints on a Chart

The following instructions assume that you are viewing the **Dashboard page** for the dashboard containing a **Chart - Category** or **Chart - Series section** for which you want to view details for a datapoint plotted on the chart.

To view details for a datapoint plotted on the chart in a Chart - Category or Chart - Series section:

- In the chart, hover over the datapoint whose details you want to view.

    The details for the datapoint appear on the chart, as illustrated in the following image that shows an example of a line chart displayed in a **Chart - Series** section.

Hiding Datapoints on a Chart

The following instructions assume that you are viewing the <u>Dashboard page</u> for the dashboard containing a <u>Chart - Category</u> or <u>Chart - Series section</u> for which you want to hide datapoints that are plotted on the chart. You cannot hide the datapoints associated with a legend item when the datapoints for that legend item are the *only* datapoints displayed on the graph (e.g., if a graph displays three legend items and you have hidden the datapoints associated with two of them, you cannot hide the datapoints associated with the remaining legend item).

To hide datapoints plotted on the chart in a Chart - Category or Chart - Series section:

- In the chart, click the corresponding item in the legend that corresponds with the bar, line, or category whose datapoints you want to hide.

  The corresponding bar, line, or category is hidden, and the legend item that you clicked turns grey in color. Additionally, for:

  - **Bar and line charts in Chart - Series sections:** The y-axis is redrawn to an appropriate scale for the remaining displayed datapoints. For example, in the following image, Recommendations in both a Created *and* Approved state are plotted by lines on the chart.

The following image shows what this same chart will look like when you hide Recommendations in an Approved state by clicking the Approved legend item.

Note: For line charts in **Chart - Series** sections, only the text for the legend item that you clicked turns grey in color, as shown in the preceding image. For bar charts in **Chart - Series** sections, the text *and* color square for the legend item turn grey in color.

- **Doughnut charts in Chart - Category sections:** The chart is redrawn to exclude the category that you have hidden. For example, in the following image, *all* criticality rankings for assets are plotted on the chart (i.e., High, Medium, Low, and Incomplete).
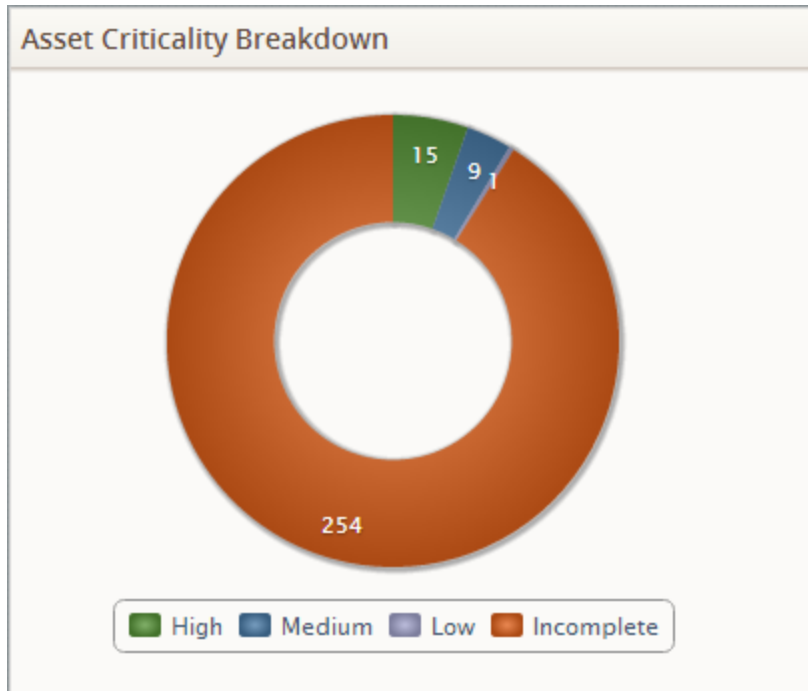
The following image shows what this same chart will look like when you hide assets with a ranking of Incomplete by clicking the Incomplete legend item.
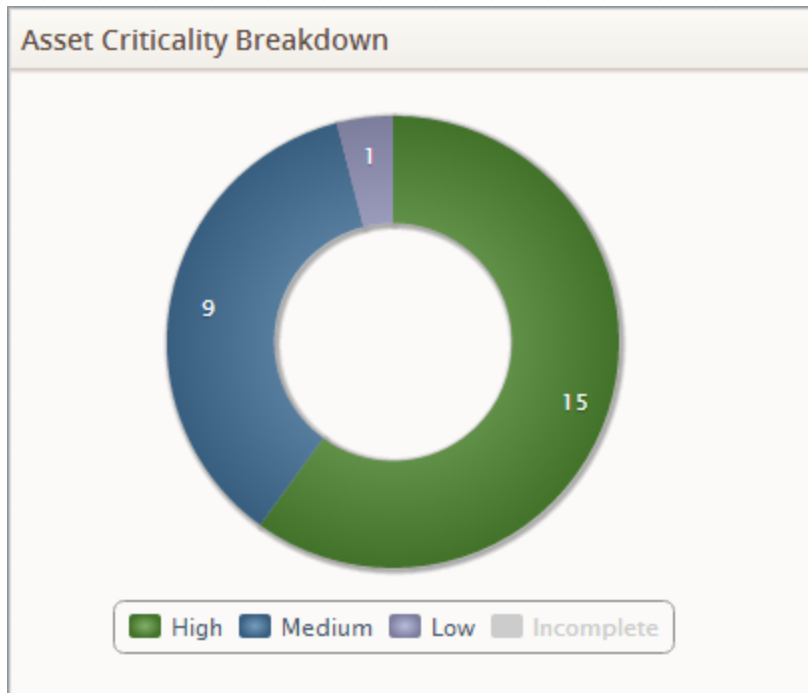
Redisplaying Hidden Datapoints on a Chart

The following instructions assume that you are viewing the Dashboard page for the dashboard containing a Chart - Category or Chart - Series section for which you want to redisplay hidden datapoints that are plotted on the chart.

To redisplay hidden datapoints on the chart in a Chart - Category or Chart - Series section:

- In the chart, click the item in the legend that corresponds with the bar, line, or category whose datapoints are hidden that you want to redisplay.

  The datapoints associated with that bar, line, or category are redisplayed, and the legend item that you clicked changes from grey to the color of the bar, line, or category that is plotted in the chart. Additionally, for:

  - **Bar and line charts in Chart - Series sections:** The y-axis is redrawn to an appropriate scale for the displayed datapoints. For example, in the following image, datapoints that are plotted by the line representing Recommendations in an Approved state are hidden.

  The following image shows what this same graph will look like when you redisplay the line representing Recommendations in an Approved state by clicking the Approved legend item.

- **Doughnut charts in Chart - Category sections:** The chart is redrawn to include the hidden datapoints. For example, in the following image, the category associated with equipment with an Incomplete criticality are hidden.
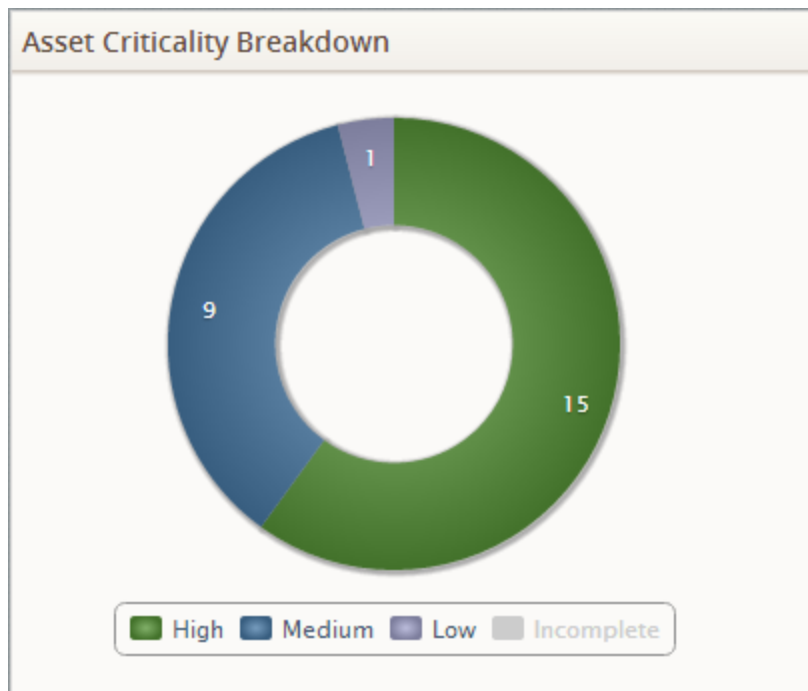
The following image shows what this same graph will look like when you redisplay the category associated with assets with an Incomplete criticality by clicking the Incomplete legend item.
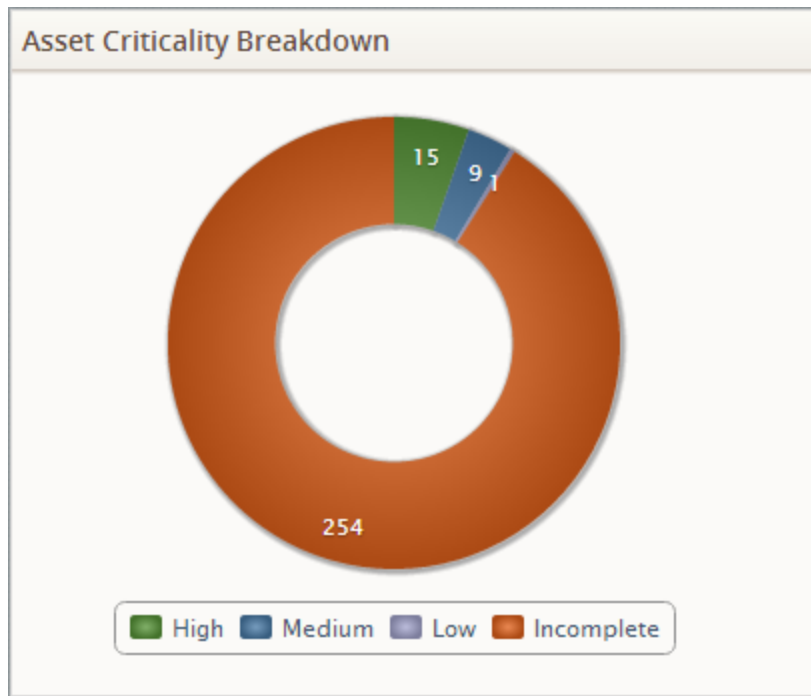
Refreshing Dashboards

If you work with a dashboard that was created by another user or if another user makes updates to the dashboard or any of its underlying queries, you should refresh the dashboard to ensure that the dashboard Catalog item and any data returned by the underlying queries are up-to-date.

To refresh a dashboard:

- On the **Dashboard** page, on the Meridium APM Framework toolbar, click the **Refresh** button.

  The dashboard is Catalog item and any data returned by underlying queries used by the dashboard are refreshed.

Sharing Dashboards via Email

When viewing a dashboard, you can send a link to the dashboard via email. The email recipient can then click the link to view the dashboard in a web browser.

> **Note:** To view the dashboards, email recipients must have access to the Meridium APM Framework application and must have a valid Meridium APM Security User account in order to log in.

To share a link to a dashboard via email message:

1. Click the [icon] button.

    The **Email Message** dialog box appears. The link is included in the message by default. The name of the data source to which you are currently logged in appears in the URL.

    > **Note:** If you are running the Meridium APM Framework application on the Application Server, the URL will contain *http://localhost/*. In this case, before you send the message, you will need to change *localhost* to the name of the computer you are currently using. An example is shown in the following image.
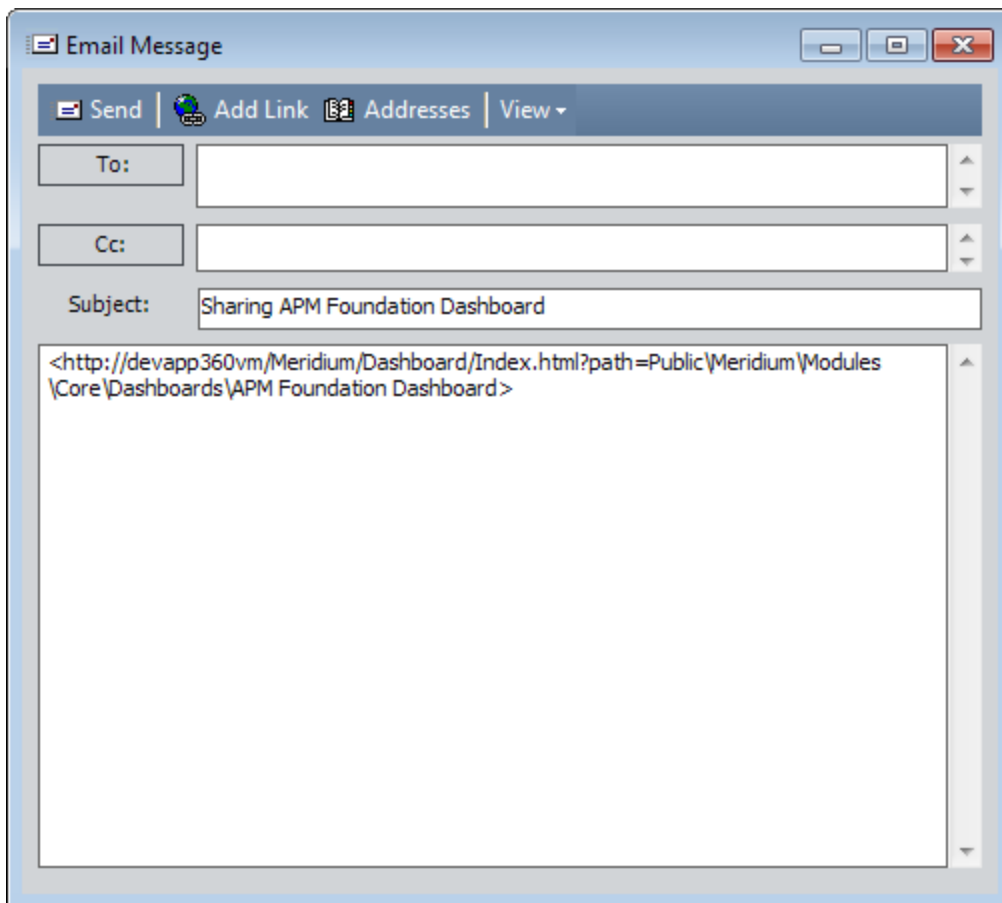
2. Send the email message.

About Dashboard Errors and Validation Messages

When you work with dashboards, you may encounter:

- [Load errors](#).
- [Validation messages](#).

This section of the documentation provides details on why you would encounter such errors or messages and how to resolve the issues related these errors and messages.

Load Errors

A load error will be displayed if you try to access the **Dashboard** page, but there is a problem with your Meridium APM Application Server (e.g, the server is down). In this case, you should contact your organization's IT department to resolve the issue with your Meridium APM Application Server.

Validation Messages

Validation messages are displayed on the **Dashboard** page when a dashboard section's underlying query and/or query parameters contain invalid values or are not defined. To resolve this issue, a user who has Edit privileges to the Catalog folder that stores the dashboard and its underlying queries should ensure that valid query paths and parameters are specified in the edit view of each section on the dashboard.

The following callouts have been added to the following image to highlight information that is displayed when validation messages are displayed:

- **1:** The text *<number of current message> of <total number of messages>* is displayed, where *<number of current message>* is the message for which details are currently displayed to the right of the **Go to Source** link, and *<total number of messages>* is the total number of validation messages displayed on the dashboard. Buttons are displayed to the left and right of this text that can be used to view the descriptions of other validation messages. In the following image, the text *1 of 2* is displayed.

- **2:** The **View Details** link is displayed, which displays the **Error Details** dialog box for the validation message.

- **3:** The **Go to Source** link is displayed, which displays the edit view of the dashboard section that is the source of the validation message.

- **4:** The text *<Dashboard section>: <Validation message>* is displayed, where *<Dashboard section>* is the section that is the source of the validation message, and *<Validation message>* is the validation message. In the following image, the text *List - Summary: No query path specified* is displayed.

- **5:** Validation messages are displayed in each section for which the underlying queries and/or query parameter(s) contain invalid values or are not defined.

Validation Messages

Options for Creating Datasets

You can create datasets using the **Dataset Builder**, where you can create a dataset in the following ways:

- Create a dataset by entering data manually.
- Create a dataset by importing a Microsoft Excel file (.XLS).
- Create a dataset by importing a text file (.TXT or .CSV).
- Create a dataset by importing a XML file (.XML).

After you have created a dataset using the **Dataset Builder**, you will be able to modify the content of the dataset (i.e., data can be changed, and rows can be added and deleted), but you cannot make changes to the column structure of the dataset. While columns can be rearranged or hidden from view within the grid on the **Dataset Editor page**, columns cannot be added to or deleted from the dataset itself. Therefore, when you create a new dataset, you want to be sure that it contains the correct column structure.

After you create a dataset using the **Dataset Builder**, you will need to save the dataset to the Catalog if you want to be able to access it later.

Creating Datasets by Entering Data Manually

One option for creating a dataset is to define the columns that you want it to include and then, via the **Dataset Builder**, enter some or all of the data that the dataset will contain. For instance, consider the [example dataset](#) and assume that instead of using a security system that logs information automatically, entries into a building are recorded manually by a security guard in a paper log. At the end of his shift, the security guard might transfer his written entries to a dataset in the Meridium APM Framework using the following procedure.

To create a dataset by entering data manually:

1. In the Meridium APM Framework, on the Meridium APM Framework toolbar, click the **Dataset** button, and then click **Build New**.

   The **Dataset Builder** appears, displaying the **Choose a Data Source** screen.
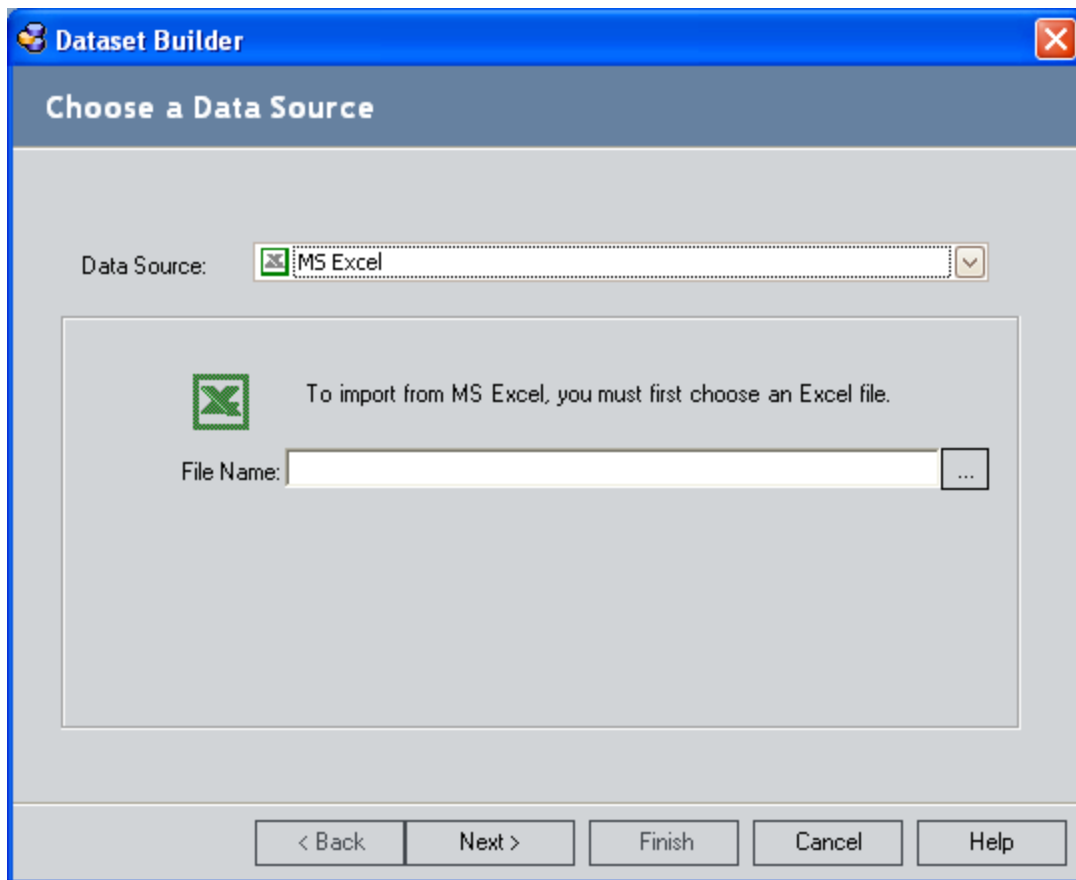
2. In the **Data Source** list, select **Design a new dataset**.

3. Click the **Next** button.

   The **Design Dataset** screen appears.
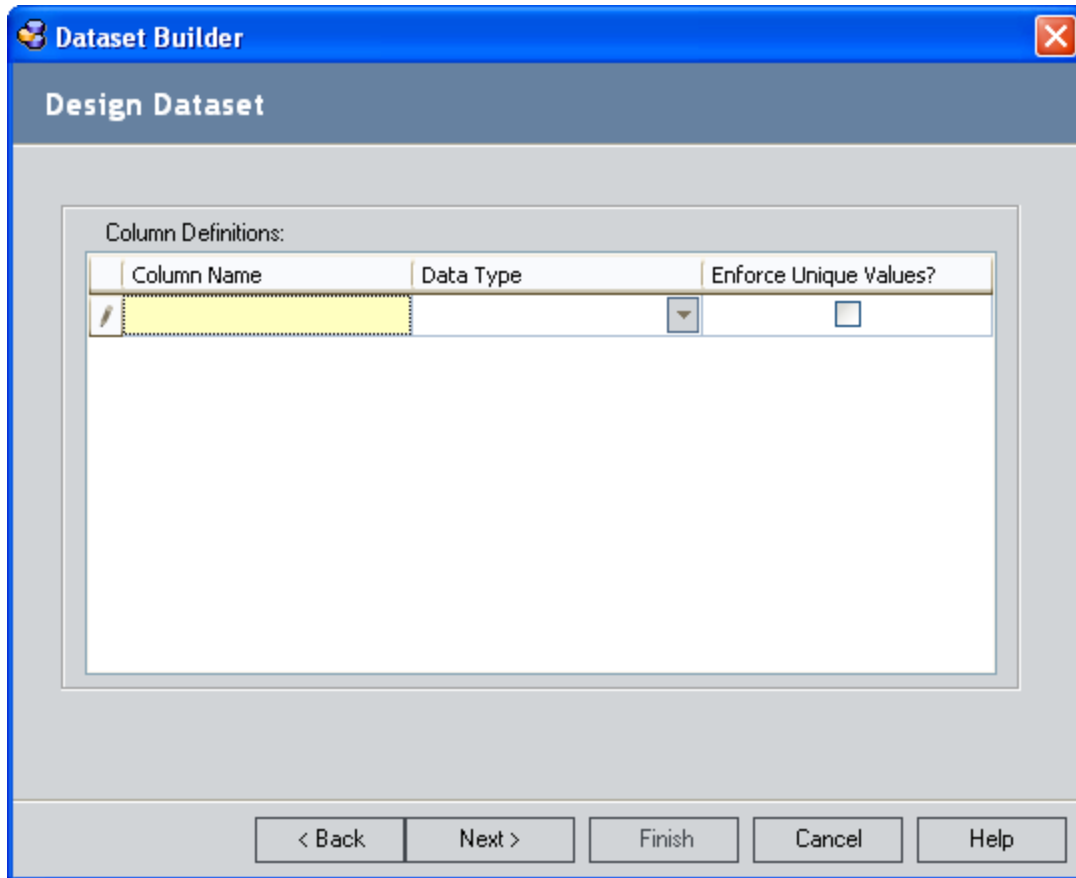
4. In the **Column Name** text box, type a name for the first column in the dataset. To create the first column in our example dataset, we would type **Entry.**

5. In the **Data Type** list, select the type of data that will be allowed in the column. For the ENTRY column, we would select **Number.**

6. If you want to ensure that every cell in the column contains a unique value with respect to other values in the same column, select the **Enforce Unique Values?** check box. For the ENTRY column, because each value should be unique, we would select this check box.

7. If you want to add another column to the dataset, position the cursor in the **Enforce Unique Values?** cell of the first row, and press the Tab key.

   A new row appears in the grid.

8. In the new row, define settings for the new column as desired.

   Note: Column names must be unique. If you type the same column name more than once, a message will appear when you click the **Next** button, and you will need to return to the **Design Dataset** screen to modify one of the column names.

9. After you have finished defining all the desired columns, click the **Next** button.

   The **Preview Data** screen appears, displaying the columns that you defined for the dataset.
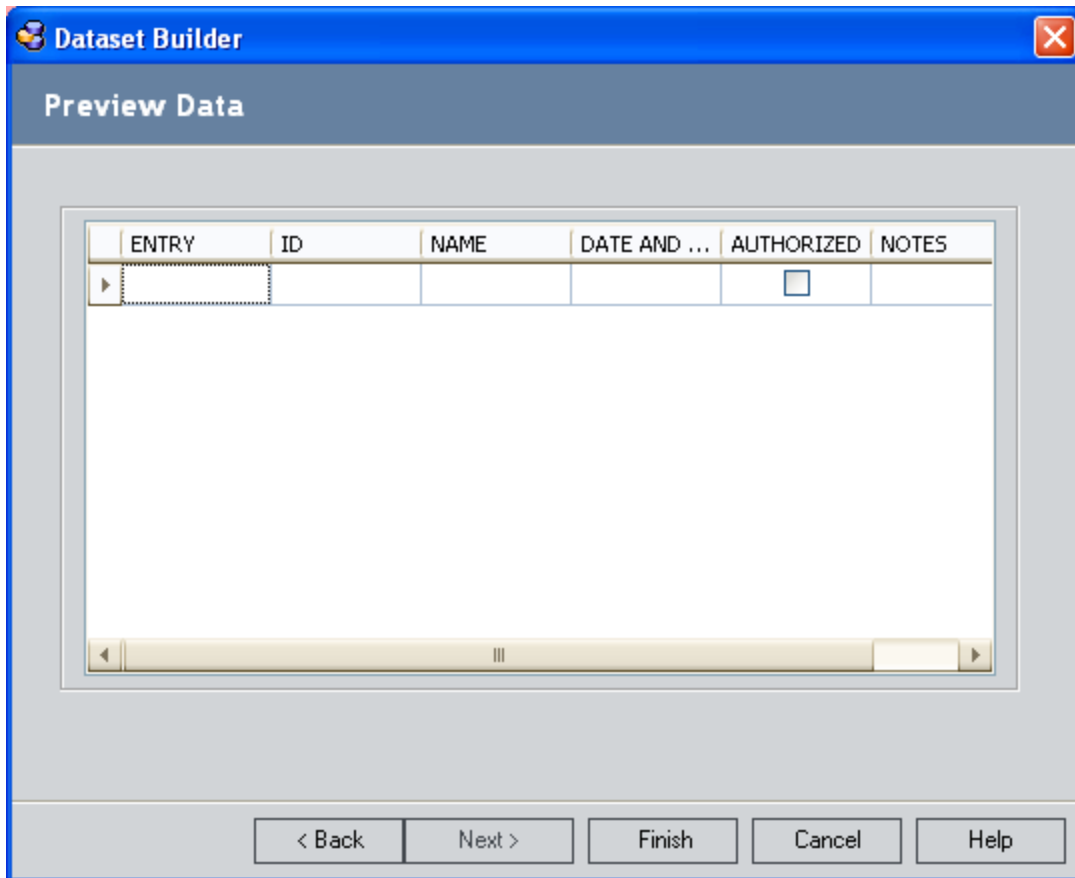
   In this image, the **Preview Data** screen shows the columns that would need to be defined to duplicate the example dataset.

10. Define values for the dataset. To do so, type a value in each empty cell. After you have typed a value, you can press the Tab or Enter key to move to the next cell. Similarly, when you reach the last column in the dataset, you can press the Tab or Enter key to create a new row. Note that each value you type must correspond to the data type that you selected for each column.

11. Click the **Finish** button.

    The dataset appears on the **Dataset Editor** page.

12. If desired, you can further modify the dataset in the **Data Editor** workspace.

13. On the **Common Tasks** menu, click the **Save this dataset** link, and save the dataset to the desired location in the Catalog.

    The dataset is saved in the Catalog.

About the Import Process

You can import a dataset from a [Microsoft Excel file](#), a [text file](#), or an [XML file](#). To import a dataset, it must be stored in some location that you can access, such as your local computer or a network drive.

After you import a dataset, you can [modify it as needed](#), save it to the Meridium APM Catalog, and use it to [create graphs](#) and [reports](#) in the Meridium APM Framework application.

About Importing Microsoft Excel Files

Microsoft Excel files with the file extension .XLS can be imported to create a dataset. In a Microsoft Excel file, data is organized within grids on worksheets, or tabs. A dataset can be created from a single worksheet only. If you want to import a file that contains multiple worksheets, you must create multiple datasets, one for each worksheet that you want to import.

In Microsoft Excel files, information is stored in columns that are identified by letters and rows that are identified by numbers. The data itself is stored in the intersecting cells, which are identified according to their location within the grid (e.g., the cell at the intersection of Column A and Row 1 is *A1*). Understanding the grid layout will be necessary if you want to create a dataset using a certain range of cells within a Microsoft Excel file.

To create a dataset that looks like our example, the Microsoft Excel file should look like the one shown in the following image.

	A	B	C	D	E	F
1	ENTRY	ID	NAME	DATE_AND_TIME	AUTHORIZED	NOTES
2	1	11	John Doe	1/10/10 7:45 AM	True	
3	2	13	Bill Doe	1/10/10 8:05 AM	False	Unauthorized Entry Code
4	3	13	Bill Doe	1/10/10 8:10 AM	True	
5	4	12	Jane Doe	1/10/10 11:15 AM	True	
6	5	13	Bill Doe	1/10/10 12:30 PM	True	
7	6	14	Sarah Doe	1/10/10 1:09 PM	True	
8	7	14	Sarah Doe	1/10/10 3:15 PM	True	
9						
10						

In this example, the first row in the Microsoft Excel file contains labels that identify the information that is stored in each column, such as ENTRY, ID, and NAME. Each subsequent row represents a single entry into the building and contains all the information related to that entry. For example, row 2 contains the information for one authorized entry by John Doe, at 7:45 A.M. on January 10, 2010.

> **Note:** If the first row in a Microsoft Excel file contains values that you want to use as column names for the dataset, as shown in the example, when you create the dataset, you should leave the **First row contains header definition?** check box selected in the **Dataset Builder**. If the first row in the Microsoft Excel file does not contain values that you want to use as column names, you should clear this check box and allow the columns to be created with default names F1, F2, F3, and so on.

When you import a Microsoft Excel file to create a dataset, the Meridium APM system will import the data and choose a data type for each column based upon the values *stored* in each column.

- Columns that store *only* numbers will be assigned the *Numeric* data type.
- Columns that store *only* date or time data will be assigned the *Date* data type.

- Columns that store letters and/or symbols will be assigned the *Character* data type. The column *may* also contain numbers, but unless the column is limited to containing *only* numbers, it will be imported using the Character data type.

- Columns that contain *only* the values *True* or *False* will be assigned the *Logical* data type.

> **Note:** In a Microsoft Excel file, the *stored* value is not necessarily the same as the displayed value. For example, a cell may *store* the value 2 but may be formatted to *display* 2%. In this case, the percent symbol is not actually stored in the cell. In Microsoft Excel, you can view the formatting options that are applied to a given cell to determine how the stored value differs from the displayed value.

If needed, on the **Modify Column Descriptions** screen of the **Dataset Builder**, you can modify any data type that is assigned incorrectly during the import process. For example, any column that contains only numbers will be interpreted as a Numeric column. If that column can contain numbers, letters, and symbols, you will want to change the data type to Character so that users can modify the data to include letters and symbols after the dataset has been created.

Additionally, a column that contains only the values 0 (zero) and 1 (one) will be assigned the Numeric data type. You can change the data type to Logical if you want to import the values as logical data.

> **Note:** If you change the data type for a column to a selection that is not appropriate for the data that has been imported, the data for that column may not be imported or may be displayed incorrectly. For example, if you change the data type for a column containing only numbers from Number to Logical, the values will be overwritten by *True* values.

If you select the **Enforce Unique Values?** check box on the **Modify Column Descriptions** screen of the **Dataset Builder** for a column that has repeated values, only the first row with a unique value will be imported. Subsequent rows containing the same value will not be imported. For instance, if you selected the **Enforce Unique Values?** check box for the NAME column in the example, only the first row of data for each person would be imported.

Creating Datasets by Importing Microsoft Excel Files

To create a dataset by importing a Microsoft Excel file:

1. In the Meridium APM Framework, on the Meridium APM Framework toolbar, click the **Dataset** button, and then click **Build New**.

   The **Dataset Builder** appears, displaying the **Choose a Data Source** screen.
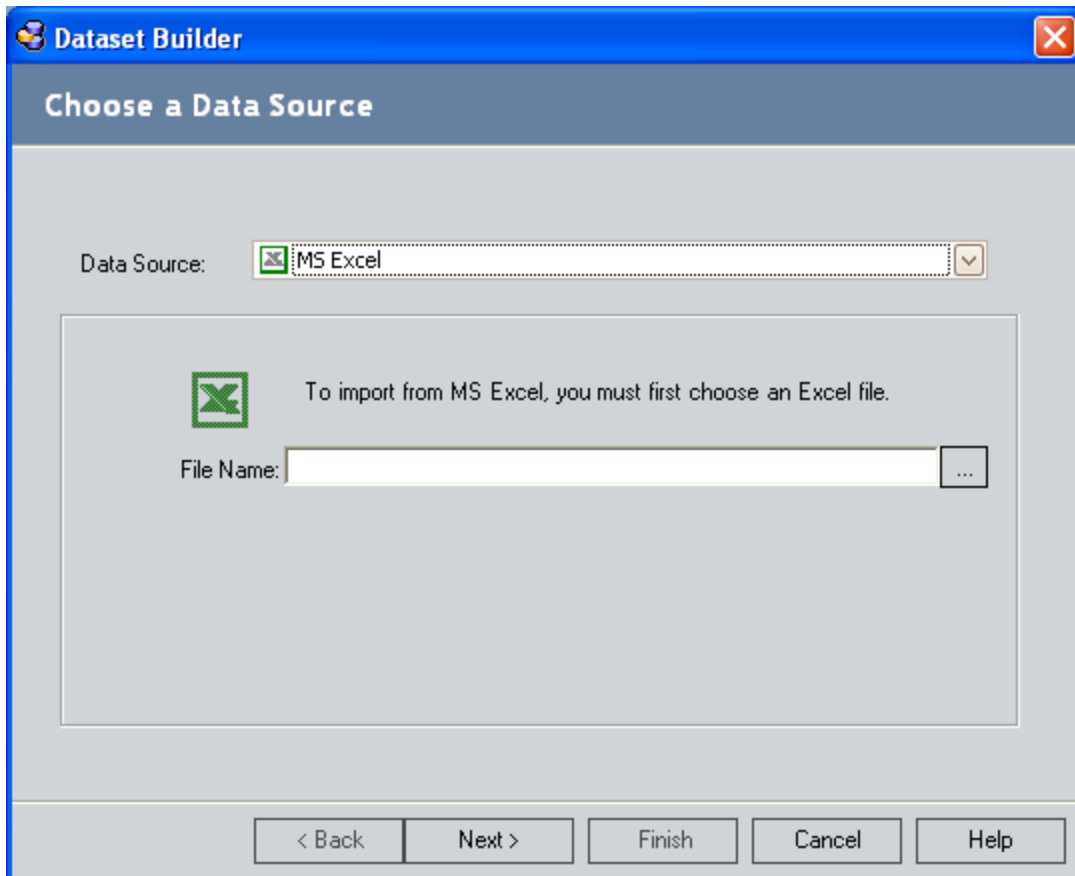
2. In the **Data Source** list, accept the default selection: **MS Excel**.

3. To the right of the **File Name** text box, click the [...] button.

   The **Open** dialog box appears.

4. Navigate to the Microsoft Excel file that you want to import, select it, and click the **Open** button.

   The path and file name of the file that you selected appears in the **File Name** text box.

5. Click the **Next** button.

The **Choose a Worksheet** screen appears.
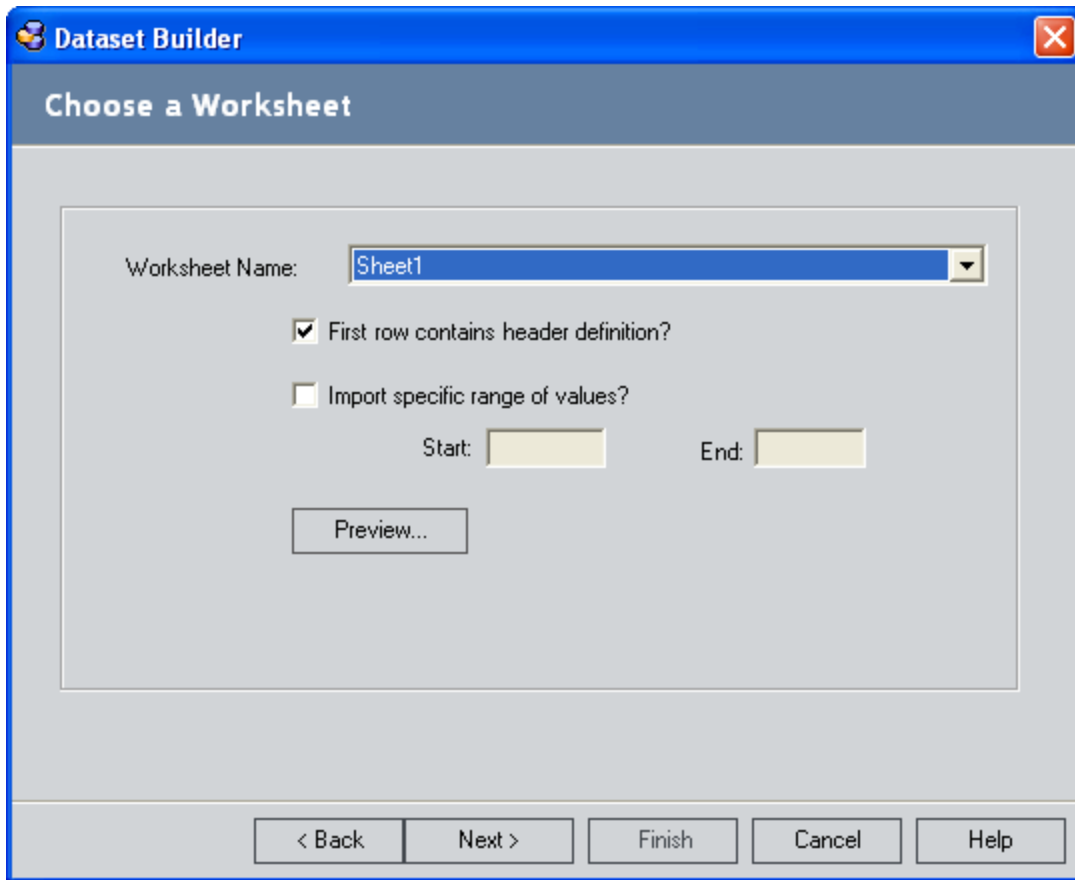
6. In the **Worksheet Name** list, select the worksheet that you want to import. The **Worksheet Name** list contains a list of all the worksheets, or tabs, that exist in the Microsoft Excel file.

7. If the first row in the worksheet should not be used as column names, clear the **First row contains header definition?** check box. For our example, we would leave this check box selected.

> Note: When the **First row contains header definition?** check box is cleared, the dataset be will created with default column names, starting with F1, F2, F3, and so on.

8. If you want to import values only from certain cells in the worksheet, select the **Import specific range of values** check box. Then in the **Start** and **End** text boxes, type the values that define the desired range (e.g., A2 to E20). For instance, using the example, if you want import the headers and only the first three rows of data, in the **Start** text box you would type **A1** and in the **End** text box you would type **F4**.

9. If desired, click the **Preview** button to see the dataset that will be created based upon your selections.

10. Click the **Next** button.
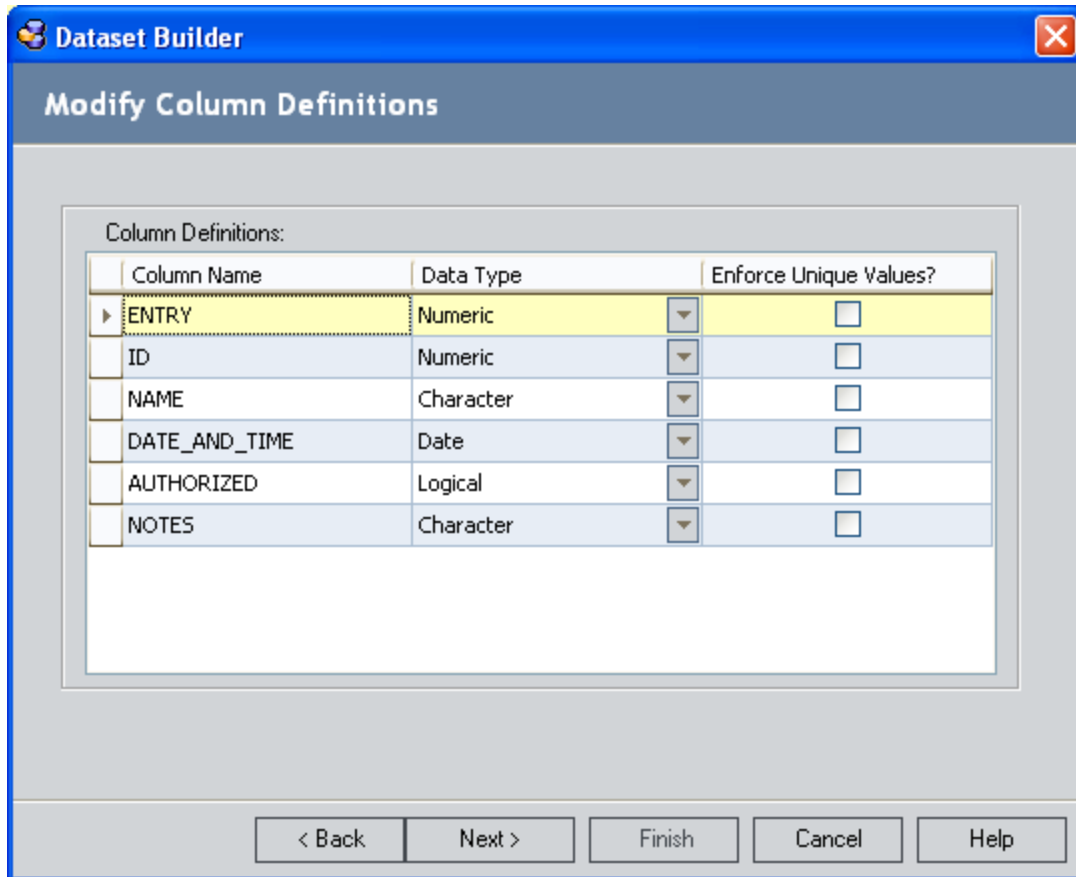
    The **Modify Column Definitions** screen appears.

    In this image, the **Modify Column Definitions** screen shows the column definitions that would be assigned automatically if you imported our example Microsoft Excel file.

11. Review the information that was imported from the Microsoft Excel file, and make any desired changes.

    a. If you want to change a column name, modify the value in the **Column Name** text box for the desired row.

    b. If you want to modify a data type, click the **Data Type** list of the desired row, and select a different data type.

12. If you want to ensure that every cell in the column contains a unique value with respect to other values in the same column, select the **Enforce Unique Values?** check box. For our example, we would select the **Enforce Unique Values?** check box for the ENTRY and DATE_AND_TIME columns.

13. Click the **Next** button.
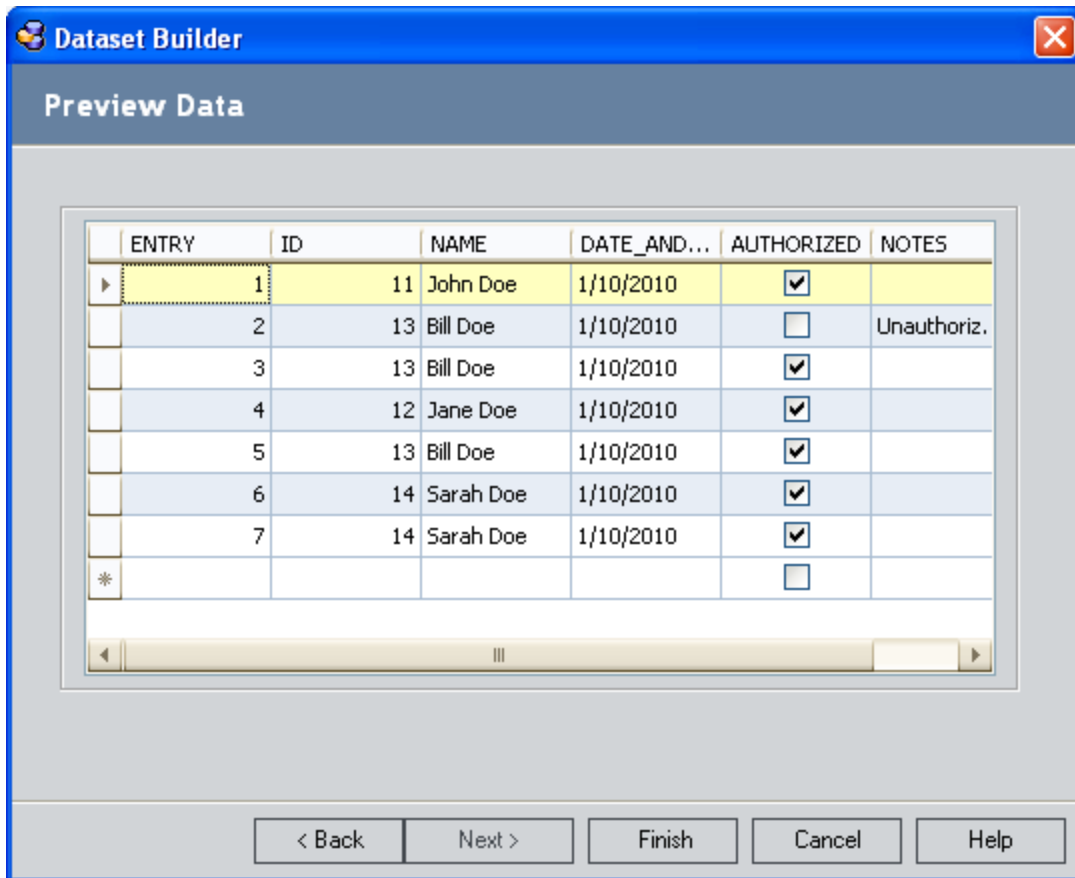
The **Preview Data** screen appears.

This image shows how the **Preview Data** screen would look after you imported our example Microsoft Excel file.

> **Note:** If a date field contains a time value, the time value will not be displayed on the **Preview Data** screen. The time will, however, be imported and will be displayed when the dataset is displayed on the Dataset Editor page.

14. Click the **Finish** button.

    The dataset appears on the **Dataset Editor** page.

15. If desired, you can further modify the dataset in the Data Editor workspace.

16. On the **Common Tasks** menu, click the **Save this dataset** link, and save the dataset to the desired location in the Catalog.

    The dataset is saved in the Catalog.

About Importing Text Files

Using the **Dataset Builder**, you can create a dataset by importing a text file that contain lines of data that are delimited or separated by punctuation marks, such as commas or semicolons, with rows of data separated by line breaks. The Meridium APM Framework supports the use of text files with the extension .TXT or .CSV.

To create a dataset that looks like our example, the text file should look like the following image.
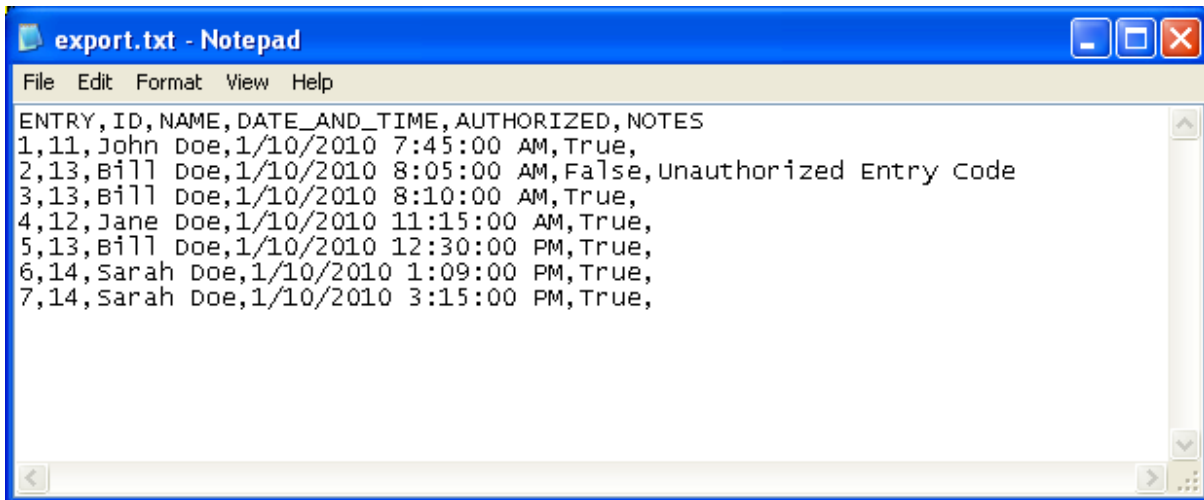
In this image, the first line contains values that will be used as column names within the dataset. Each subsequent line of text represents one row within the dataset. For example, the second line of data in the file represents an authorized entry by John Doe, at 7:45 A.M. on January 10, 2010.

> **Note:** If the first row of data in a text file contains values that you want to use as column names for the dataset, as shown in the example, when you create the dataset, you should leave the **First row contains header definition?** check box selected in the **Dataset Builder**. If the first row in the text file does not contain values that you want to use as column names, you should clear this check box and allow the columns to be created with default names F1, F2, F3, and so on.

Also in this image, commas are used to delimit, or separate, the values in the file. The Meridium APM Framework accepts the following characters as delimiters:

- **,** (comma).
- **;** (semicolon).
- **|** (pipe character).

The delimiters within the text file create the column structure. For instance, when our example text file is imported to create a dataset and a comma is identified as the delimiter, the Meridium APM system will recognize that the first column header is ENTRY and the second column header is ID because the two values are separated by a comma. On

the second row, the Meridium APM system would understand that the value 1 belongs in the ENTRY column and the value 11 belongs in the ID column because the two values are also separated by a comma.

To maintain the structure of the dataset, delimiters must be used throughout the file, even for cells that do not contain values. To identify an empty cell, you would omit a value for that column but include the delimiter that separates it from values in other columns. For example, if the last entry in the file did not have an ID value associated with it, it would look like this:

   7,,Sara Doe, 1/10/2010 3:15 PM, True,

The **Dataset Builder** imports every column as a Character data type by default. You can change this selection in the **Data Type** list on the **Modify Column Definitions** screen of the **Dataset Builder**.

> **Note:** If you try to change the data type to a selection that is not appropriate for the data that has been imported, the data for that column may not be imported or may be displayed incorrectly. For example, if you change the data type from Character to Logical in the NAME column, the values will not be imported, and a cleared check box will be displayed for all the entries in the NAME column.

You can also change the **Enforce Unique Values?** check box selection in the **Modify Column Definitions** screen. If you select the **Enforce Unique Values?** check box in a column that has repeated values, subsequent rows containing the same value will not be imported. For instance, if you selected the **Enforce Unique Values?** check box for the NAME column in our example, only the first row of data for each person would be imported.

Creating Datasets by Importing Text Files

To create a dataset by importing a text file:

1. In the Meridium APM Framework, on the Meridium APM Framework toolbar, click the **Dataset** button, and then click **Build New**.

   The **Dataset Builder** appears, displaying the **Choose a Data Source** screen.
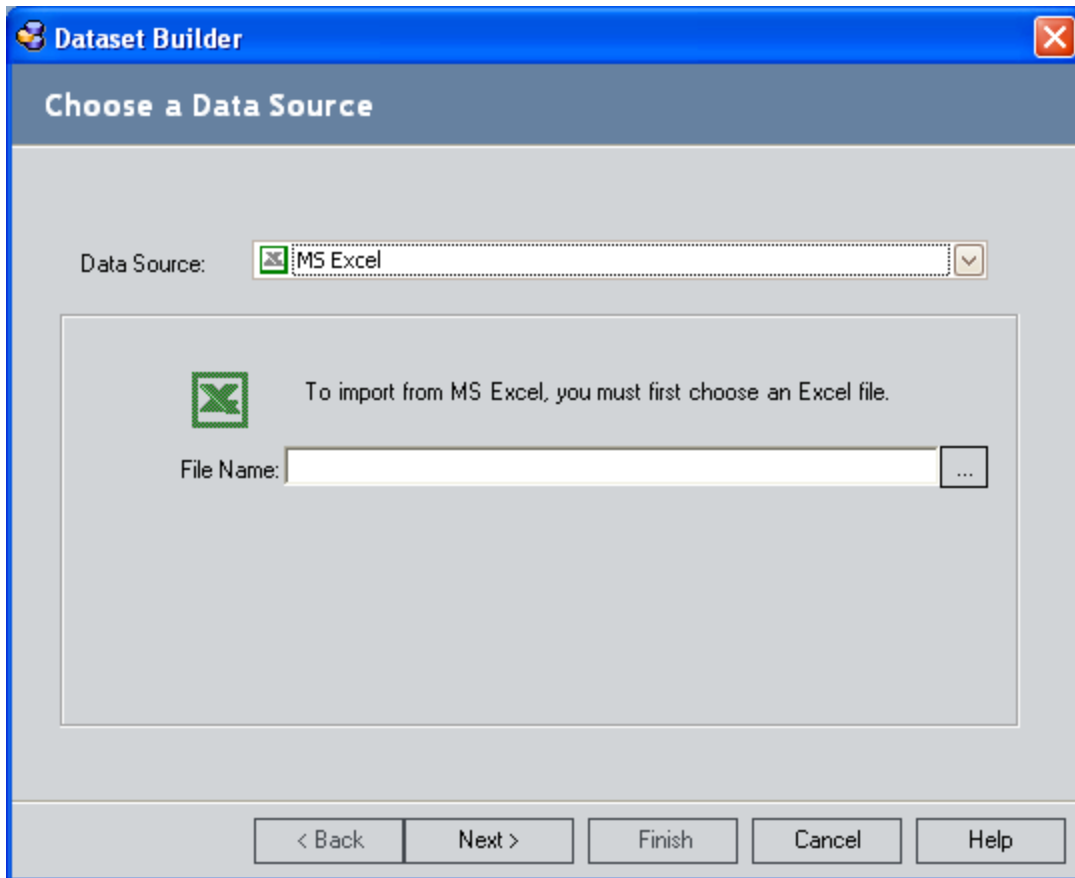
2. In the **Data Source** list, select **Text**.

3. To the right of the **File Name** text box, click the **...** button.

   The **Open** dialog box appears.

4. Navigate to the text file that you want to import, select it, and click the **Open** button.

   The path and file name of the file that you selected appears in the **File Name** text box.

5. Click the **Next** button.

   The **Choose a Column Delimiter** screen appears.
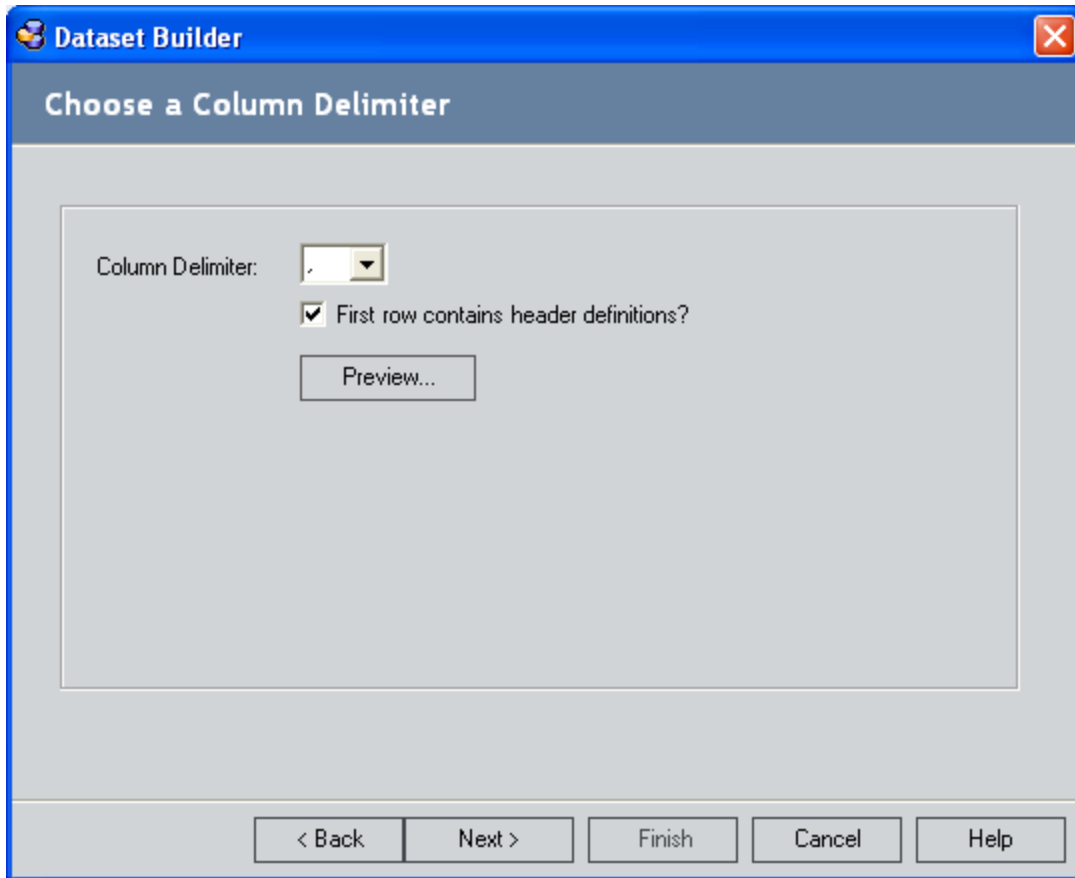
6. In the **Column Delimiter** list, select the delimiter that separates values within the text file. For our example, we would accept the default selection: **,**(comma).

7. If the first row of data in the text file should not be used as column names, clear the **First row has contains header definitions?** check box. For our example, we would leave this check box selected.

8. If desired, click the **Preview** button to see the dataset that will be created based upon your selections.

9. Click the **Next** button.
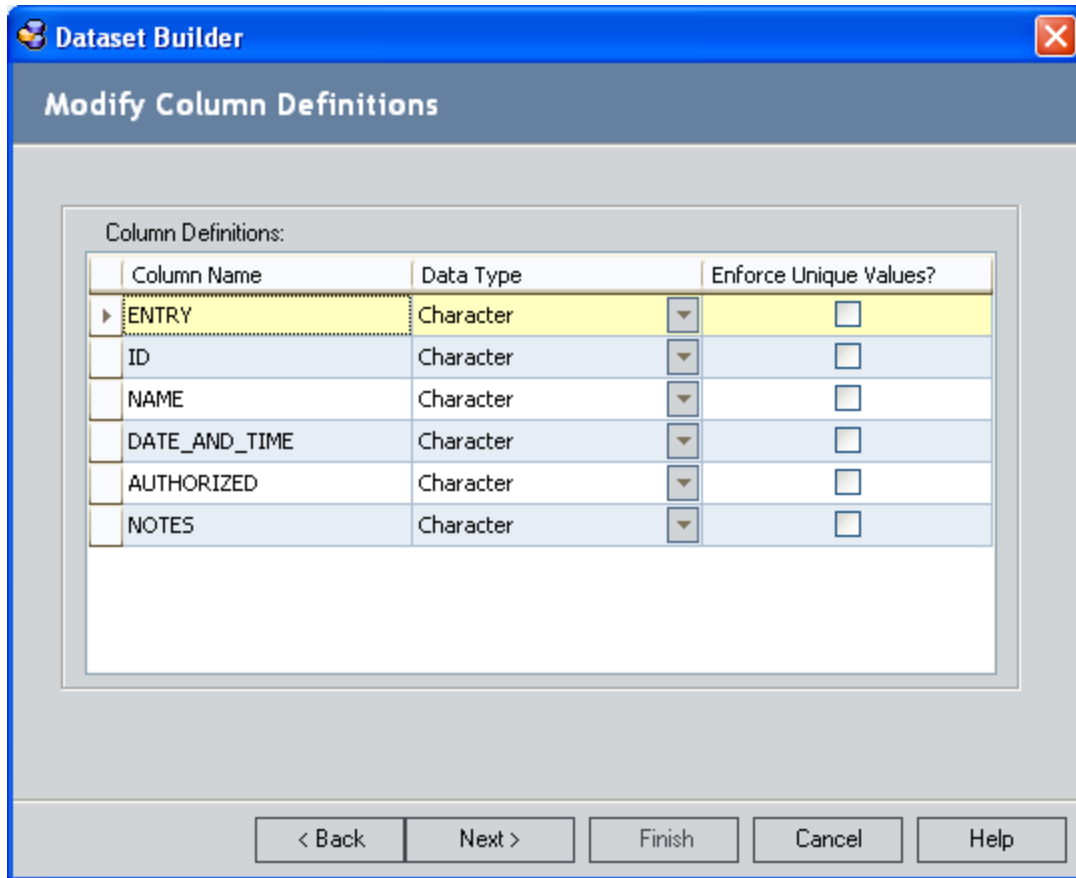
    The **Modify Column Definitions** screen appears.

In this image, the **Modify Column Definitions** screen shows the column definitions that would be assigned automatically if you imported our example text file.

10. Review the information that was imported from the text file, and make any desired changes.

   a. If you want to change a column name, modify the value in the **Column Name** text box for the desired row.

   b. If you want to modify a data type, click the **Data Type** list of the desired row, and select a different data type. For our example, we would change:

      • The ENTRY column to Numeric.

      • The ID column to Numeric.

      • The DATE_AND_TIME column to Date.

      • The AUTHORIZED column to Logical.

11. If you want to ensure that every cell in the column contains a unique value with respect to other values in the same column, select the **Enforce Unique Values?** check box. For our example, we would select the **Enforce Unique Values?** check box for the ENTRY and DATE_AND_TIME columns.

12. Click the **Next** button.
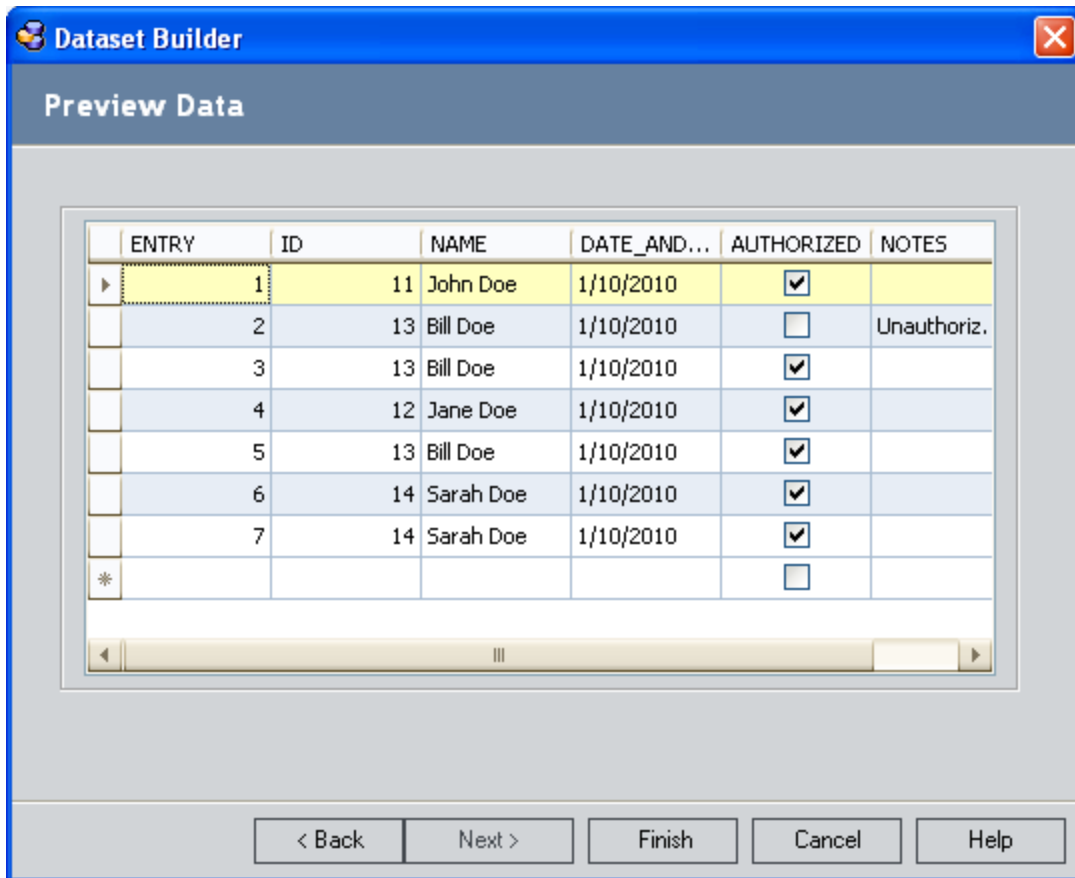
The **Preview Data** screen appears.

This image shows how the **Preview Data** screen would look after you imported our example text file.

> **Note:** If a date field contains a time value, the time value will not be displayed on the **Preview Data** screen. The time will, however, be imported and will be displayed when the dataset is displayed on the **Dataset Editor** page.

13. Click the **Finish** button.

    The dataset appears on the **Dataset Editor** page.

14. If desired, you can further modify the dataset in the **Data Editor** workspace.

15. On the **Common Tasks** menu, click the **Save this dataset** link, and save the dataset to the desired location in the Catalog.

    The dataset is saved in the Catalog.

About Importing XML Files

Extensible Markup Language (XML) files that have the file extension .XML can be imported to create Meridium APM datasets. XML files can be created in text editors, such as Notepad and Microsoft Word, but can also be opened and viewed in Internet Explorer.

To create a dataset that looks like our example, the XML file should look like the following image, which shows the file opened in Internet Explorer.
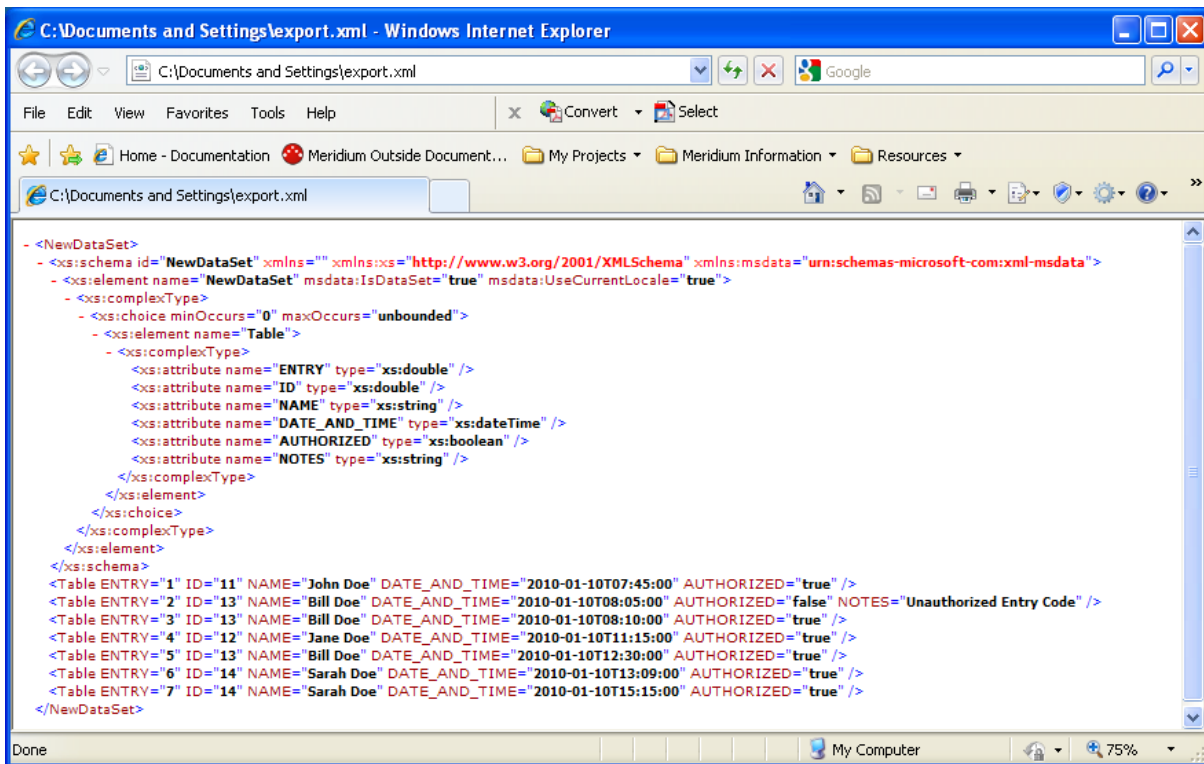
In this example, column names are identified within the **xs:attribute** tags, where the **name** attribute identifies the name of the column and the **type** attribute identifies the data type of the column. The following data types are supported:

- **double:** Used for columns that contain numeric values and may include a - (negative sign) and decimal. Columns of this type will be imported using the *Numeric* data type.

- **string:** Used for columns that contain alphanumeric values. Columns of this type will be imported using the *Character* data type.

- **dateTime:** Used for columns that contain date and time values. Date values should use the format YYYY-MM-DD, and time values should use the 24-hour time format THH:MM:SS (e.g., T14:30:23). Columns of this type will be imported using the *Date* data type.

- **boolean:** Used for columns that contain logical data. Data in these columns should be limited to the following values: True, 1, False, and 0. Columns of this type will be imported as a *Logical* data type.

If needed, on the **Modify Column Descriptions** screen of the **Dataset Builder**, you can modify any data type that is assigned incorrectly during the import process.

> **Note:** If you change the data type for a column to a selection that is not appropriate for the data that has been imported, the data for that column may not be imported or may be displayed incorrectly. For example, if you change the data type for a column containing only numbers from Numeric to Logical, the values will be overwritten by *True* values.

Each row of the dataset is contained with a **Table** tag. The attributes within the **Table** tag correspond to the column names defined via the **xs:attribute** tags. For example, in the image shown above, the first **xs:attribute** tag identifies the ENTRY column. In the first **Table** tag, the ENTRY attribute is set to 1. This means that 1 will be the value for the ENTRY column in the first row of the dataset. Empty cells are identified by an omission of the corresponding attribute within the appropriate **Table** tag.

Creating Datasets by Importing XML Files

To create a dataset by importing an XML file:

1. In the Meridium APM Framework, on the Meridium APM Framework toolbar, click the **Dataset** button, and then click **Build New**.

   The **Dataset Builder** appears, displaying the **Choose a Data Source** screen.
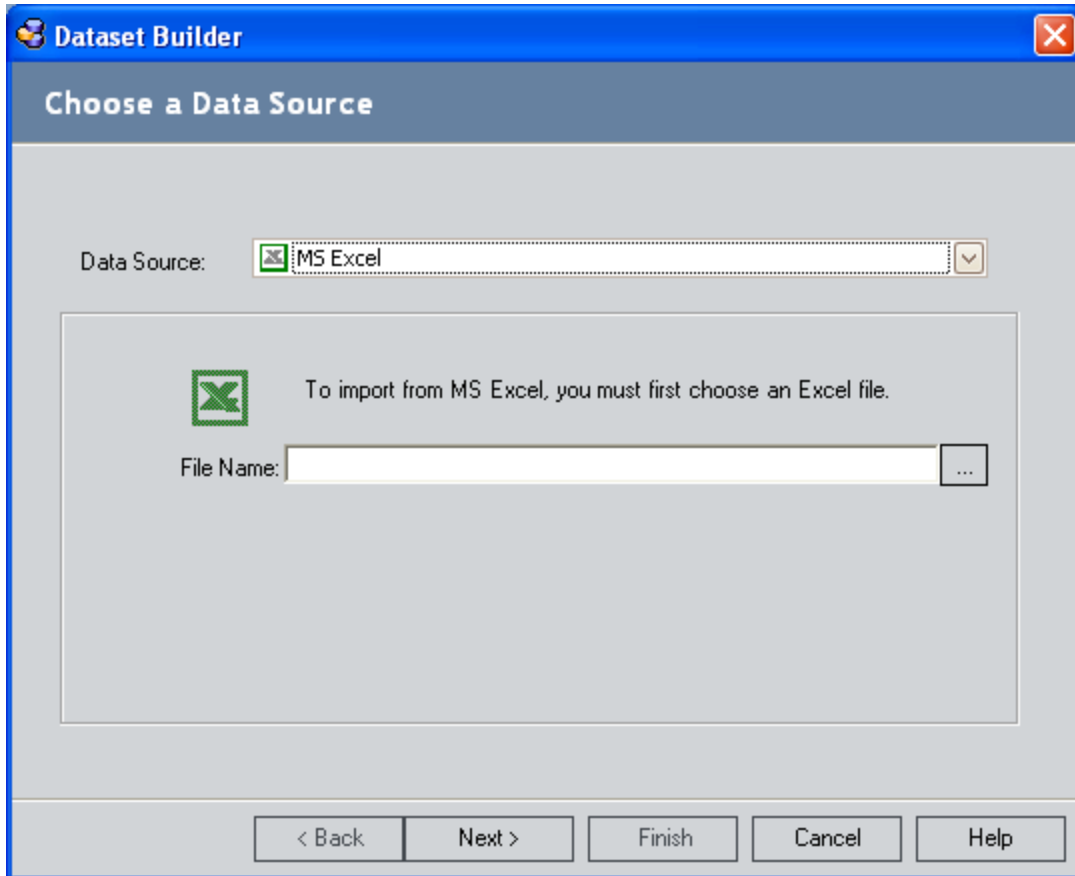
2. In the **Data Source** list, select **XML**.

3. To the right of the **File Name** text box, click the ... button.

   The **Open** dialog box appears.

4. Navigate to the XML file that you want to import, select it, and click the **Open** button.

   The path and file name of the file that you selected appears in the **File Name** text box.

5. Click the **Next** button.

   The **Modify Column Definitions** screen appears.
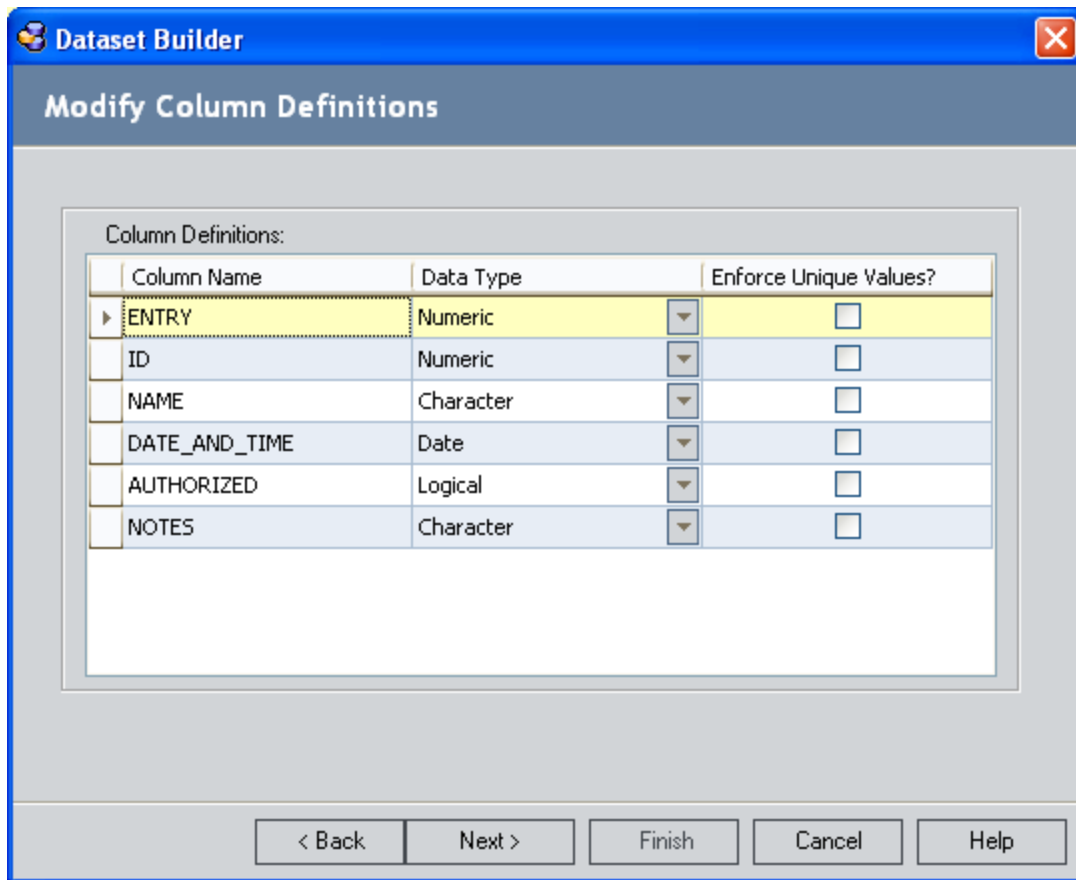
In this image, the Modify Column Definitions screen shows the column definitions that would be assigned automatically if you imported our example XML file.

6. Review the information that was imported from the XML file, and make any desired changes.

    a. If you want to change a column name, modify the value in the **Column Name** text box for the desired row.

    b. If you want to modify a data type, click the **Data Type** list of the desired row, and select a different data type.

7. If you want to ensure that every cell in the column contains a unique value with respect to other values in the same column, select the **Enforce Unique Values?** check box. For our example, we would select the **Enforce Unique Values?** check box for the ENTRY and DATE_AND_TIME columns.

8. Click the **Next** button.
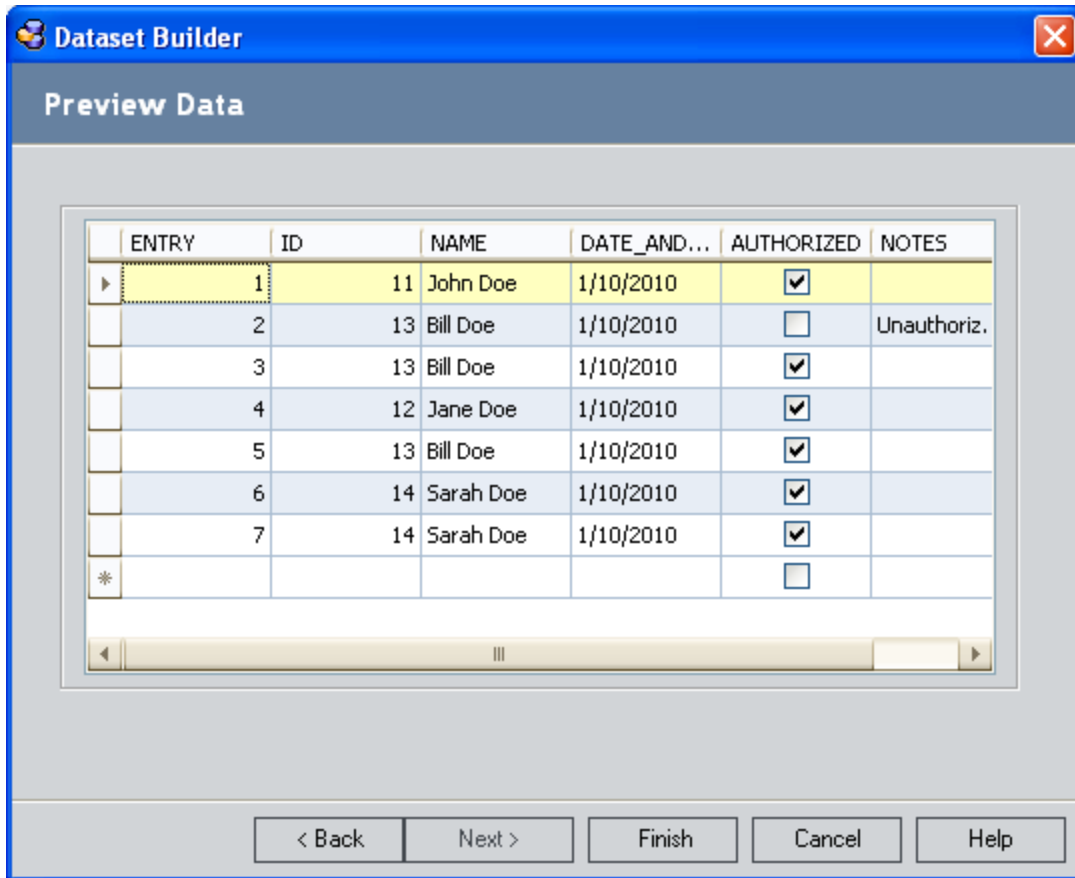
    The **Preview Data** screen appears.

This image shows how the **Preview Data** screen would look after you imported our example XML file.

> **Note:** If a date field contains a time value, the time value will not be displayed on the **Preview Data** screen. The time will, however, be imported and will be displayed when the dataset is displayed on the **Dataset Editor** page.

9. Click the **Finish** button.

   The dataset appears on the **Data Editor** page.

10. If desired, you can further modify the dataset in the **Data Editor** workspace.

11. On the **Common Tasks** menu, click the **Save this dataset** link, and save the dataset to the desired location in the Catalog.

    The dataset is saved in the Catalog.

Opening Existing Datasets

To open an existing dataset:

1. In the Meridium APM Framework, on the Meridium APM Framework toolbar, click the **Dataset** button, and then click **Open Existing**.

   The **Open an existing Dataset** dialog box appears.
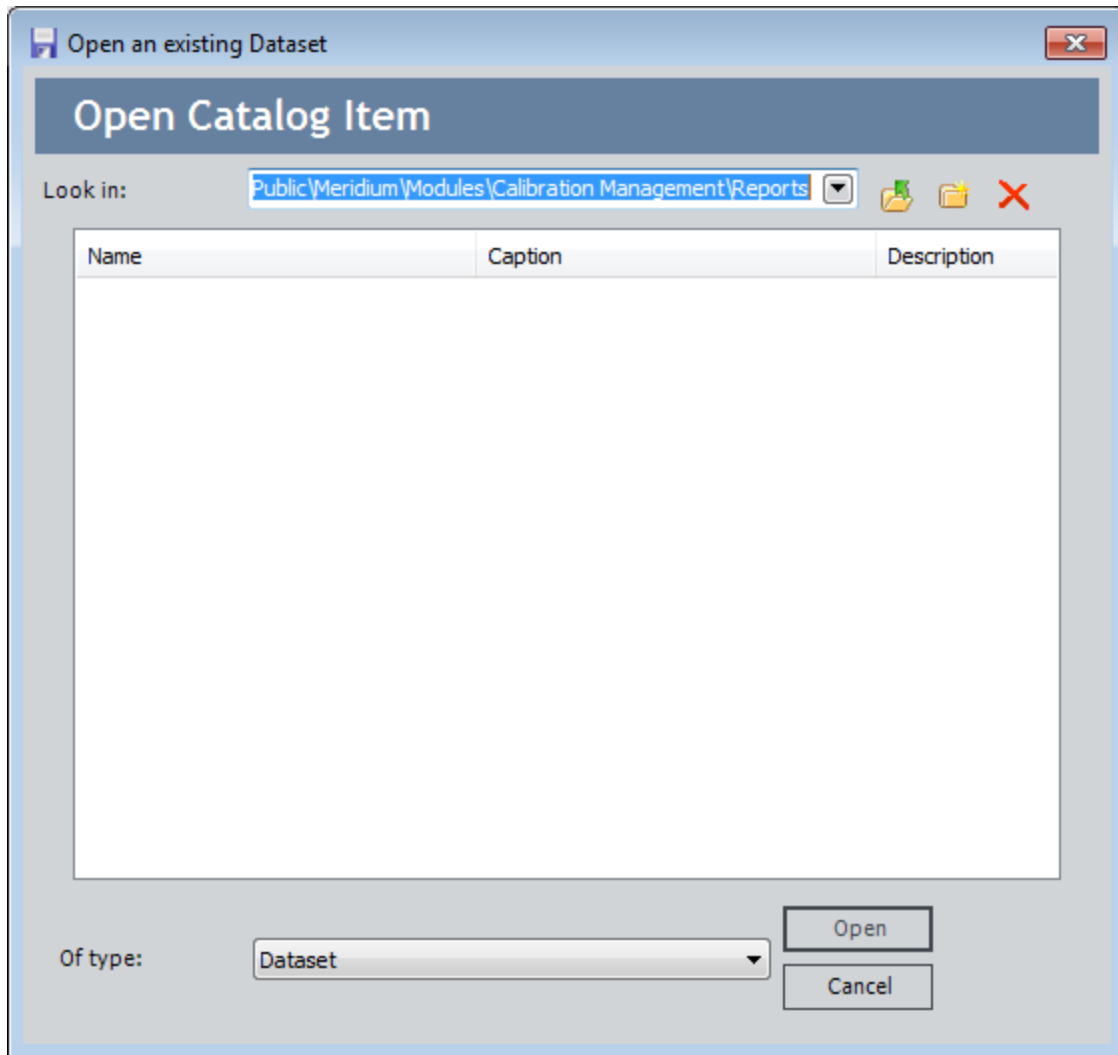
2. Navigate to the folder that contains the dataset that you want to open, select the dataset, and click the **Open** button.

   The dataset appears on the **Dataset Editor page**.

Overview of the Dataset Editor Page

The **Dataset Editor** page serves as the main user interface for working with existing data-sets and accessing functions associated with them. The **Dataset Editor** page also appears immediately after you create or import a dataset so that you can modify its content, if desired, and then save it to the Catalog.

The **Dataset Editor** page is divided into two main menus:

- Dataset Editor workspace: Contains a grid that displays the content of a given data-set.
- **Task menu pane:** Contains the **Common Tasks** and the **Other Tasks menus**, which provide access to features and functions associated with datasets.

Dataset Editor Workspace

The grid in the **Dataset Editor** workspace contains columns that organize data vertically to define the structure of the dataset. You can specify names for columns when datasets are created.

The grid also contains rows that organize data horizontally. The number of rows that appear will depend upon the amount of data that exists in the dataset. In our example dataset, a row represents one entry in the security system log, with corresponding values in each column for that instance. At the bottom of the last populated row, a blank row exists. You can type values into the empty row to add a row of data to the dataset.

If desired, you can customize the appearance of the grid (e.g., by grouping or filtering data).

> **Note:** Changes that you make to the appearance of the grid *will not be saved* to the dataset. For example, hiding a column does not delete it from the dataset. It simply removes the column from the display.

Common Tasks Menu
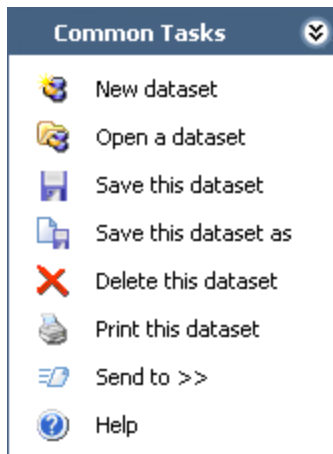
The **Common Tasks** menu on the **Dataset Editor** page contains the following links:

- **New dataset:** Displays the **Dataset Builder**, where you can create datasets by manually entering data or importing data from a Microsoft Excel file (.XLS), a text file (.TXT or .CSV), or an XML file (.XML).

- **Open a dataset:** Displays the **Open an existing Dataset** dialog box, where you can open a saved dataset.

- **Save this dataset:** Displays the **Save dataset** dialog box, where you can save the current dataset. If you are viewing a dataset that was previously saved and has been modified, clicking this link will save any changes to the existing dataset.

- **Save this dataset as:** Displays the **Save dataset** dialog box, where you can save an existing dataset with a different name or to a different location.

- **Delete this dataset:** Displays a confirmation message, asking if you want to permanently delete the current dataset. This link is enabled only when you are viewing a saved dataset.

- **Print this dataset:** Displays the saved dataset in the **Preview** window, where you can print the dataset. This link is enabled only when you are viewing a saved dataset.

- **Send to:** Displays a submenu with options that let you provide a link to the current page on your desktop (create shortcut), in an email message, or a Home Page. This link is enabled only when you are viewing a saved dataset.

- **Help:** Displays the context-sensitive Help topic for the current page.

Other Tasks Menu
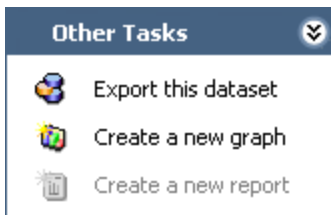
The **Other Tasks** menu on the **Dataset Editor** page contains the following links:

- **Export this dataset:** Displays the **Dataset Export Builder**, where you can export the current dataset to an external file.
- **Create a new graph:** Displays the **Graph Setup** window, where you can create a graph based upon the current dataset.
- **Create a new report:** Lets you create a report based on the dataset that you are viewing. This link is enabled only if the Microsoft SQL Server Business Intelligence Development Studio is installed on the Meridium APM Framework workstation.

Note: The links on the **Other Tasks** menu are enabled only if you are viewing a dataset that has been saved to the Meridium APM Catalog.

Options for Modifying Datasets

While datasets contain static information, you may need to update some of the data within a dataset or add additional data to it. From the **Dataset Editor** page, you can modify datasets in the following ways:

- Modify existing data.
- Add rows.
- Delete rows.

After a dataset exists, you can modify its content but not its structure, meaning that you can add, modify, and delete data but not add or delete columns. If you need to modify the column structure or the data type of a column, you can perform the following steps:

1. Export the dataset to a file.

2. Modify the file as needed to add columns, remove columns, or change data types.

3. Import the updated file to create a new dataset.

Modifying Existing Data

To modify existing data:

1. In the Meridium APM Framework, open the dataset that you want to modify.

   The dataset appears on the **[Dataset Editor](#) page**.

2. Click the cell that you want to modify.

3. Modify the text by entering new text, modifying existing text, or deleting existing text.

4. Modify the data in other cells as desired.

5. On the **Common Tasks** menu, click the **Save this dataset** link.

   Your changes are saved.

Adding Rows to Datasets

To add a row to a dataset:

1. In the Meridium APM Framework, open the dataset to which you want to add a row.

    The dataset appears on the **[Dataset Editor](#) page**.

2. Place your cursor in the blank row at the bottom of the page.

3. Enter the desired values within the new row.

4. After you have entered the desired values, click outside of the new row, or press the Tab key with your cursor positioned in the last cell of the new row.

    A new blank row is created.

5. Continue adding rows in this way until you have added all the desired rows.

6. On the **Common Tasks** menu, click the **Save this dataset** link.

    Your changes are saved.

Deleting Rows from Datasets

To delete one or more rows from a dataset:

1. In the Meridium APM Framework, open the dataset from which you want to delete one or more rows.

   The dataset appears on the **[Dataset Editor](#) page**.

2. Highlight the row that you want to delete.

3. Press the Delete key on your keyboard.

   A message appears, asking if you really want to delete the selected rows.

4. Click the **Yes** button.

   The rows are deleted from the dataset.

5. On the **Common Tasks** menu, click the **Save this dataset** link.

   Your changes are saved.

Creating Graphs from Datasets

To create a new graph from a dataset:

1. In the Meridium APM Framework, open the dataset that you want to use for creating a graph.

   The dataset appears on the **Dataset Editor** page.

2. Select options as needed to create the desired type of graph.

3. Click **OK**.

   The graph appears on the **Graph Viewer** page.

Creating Reports from Datasets

To create a report from a dataset:

1. In the Meridium APM Framework, open the dataset that you want to use for creating a report.

   The dataset appears on the **Dataset Editor** page.

2. On the **Other Tasks** menu, click the **Create a new report** link.

3. Complete the necessary steps to create a report.

About Exporting Datasets

Datasets can be exported to create files that are external to the Meridium APM system. When you export a dataset, the Meridium APM system creates a copy of the dataset, including the column structure and data, in a different format. The original dataset that is saved in the Catalog is not modified in any way and will continue to reside within the Catalog.

You can export datasets in the following formats:

- **MS Excel:** Exports the dataset to a Microsoft Excel (.XLS) file.
- **Text (Comma Separated Values):** Exports the dataset to a text file (.TXT) with a comma delimiter.

  > **Note:** While the Meridium APM Framework can *import* text files with delimiters of **,** (comma), **;** (semicolon), or **|** (pipe), only a comma can be used for *exporting* a dataset to a text file.

- **XML:** Exports the dataset to an XML file (.XML).

The images displayed in the examples of importing Microsoft files, text files, and XML files represent the files that would be created if we exported our example dataset to these file formats.

> **Note:** When you export a dataset, if a column name includes any character other than a number or letter, such as a punctuation mark or space, that character will be replaced with an underscore (_) in the export file.

After you export a dataset, you can use or modify the datasets in Microsoft Excel, a text editor, an XML editor, or any other program that can support the file format. You can also modify the dataset and import the dataset back into the Meridium APM system if there are any structural changes that need to be made to the dataset.

Exporting Datasets to External Files

To export a dataset to an external file:

1. In the Meridium APM Framework, open the dataset that you want to export.

   The dataset appears on the **Dataset Editor** page.

2. On the **Dataset Editor** page, on the **Other Tasks menu**, click the **Export this dataset** link.

   The **Dataset Export Builder** appears.

3. In the **Dataset Source** list, select the type of file that will be created from the dataset.

4. Click the **Finish** button.

   The **Export to file** dialog box appears.
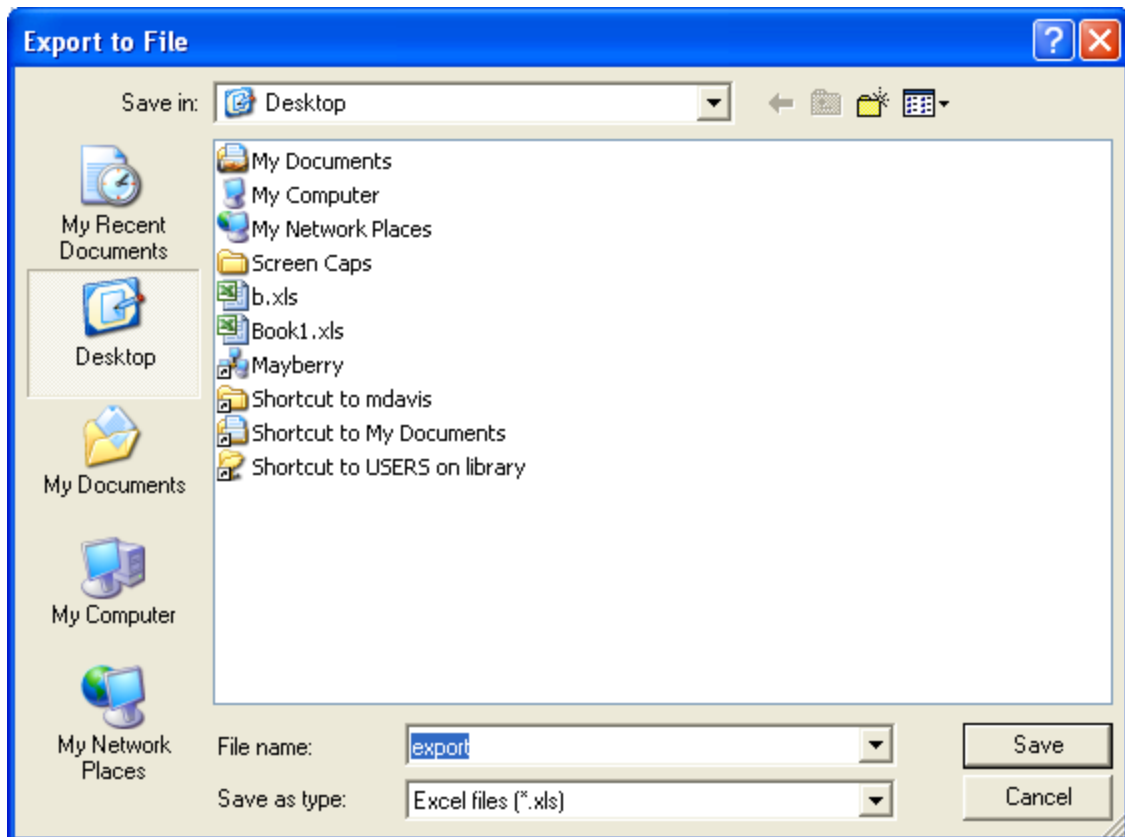
5. Navigate to the location where you want to save the exported file.

6. If desired, in the **File Name** text box, modify the file name, which is set to *export* by default.

7. Click the **Save** button.

   The dataset is exported and saved in the selected format and location.

Saving Datasets

After you create a dataset, you can save it so that you or other users can access it again later. Datasets are saved as Catalog items to the Meridium APM Catalog. After a dataset has been saved to the Meridium APM Catalog, whenever you make changes to it, you will save to re-save the dataset to retain your changes.

To save a dataset to the Meridium APM Catalog:

1. While viewing the dataset on the **Dataset Editor** page, on the **Common Tasks** menu, click the **Save this dataset** link.

   If the dataset has already been saved, any changes that you have made will be saved to the current Catalog folder and with the existing Catalog item properties. If the dataset has not yet been saved, the **Save dataset** dialog box will appear, as shown below. The following instructions assume that you are saving the dataset for the first time.
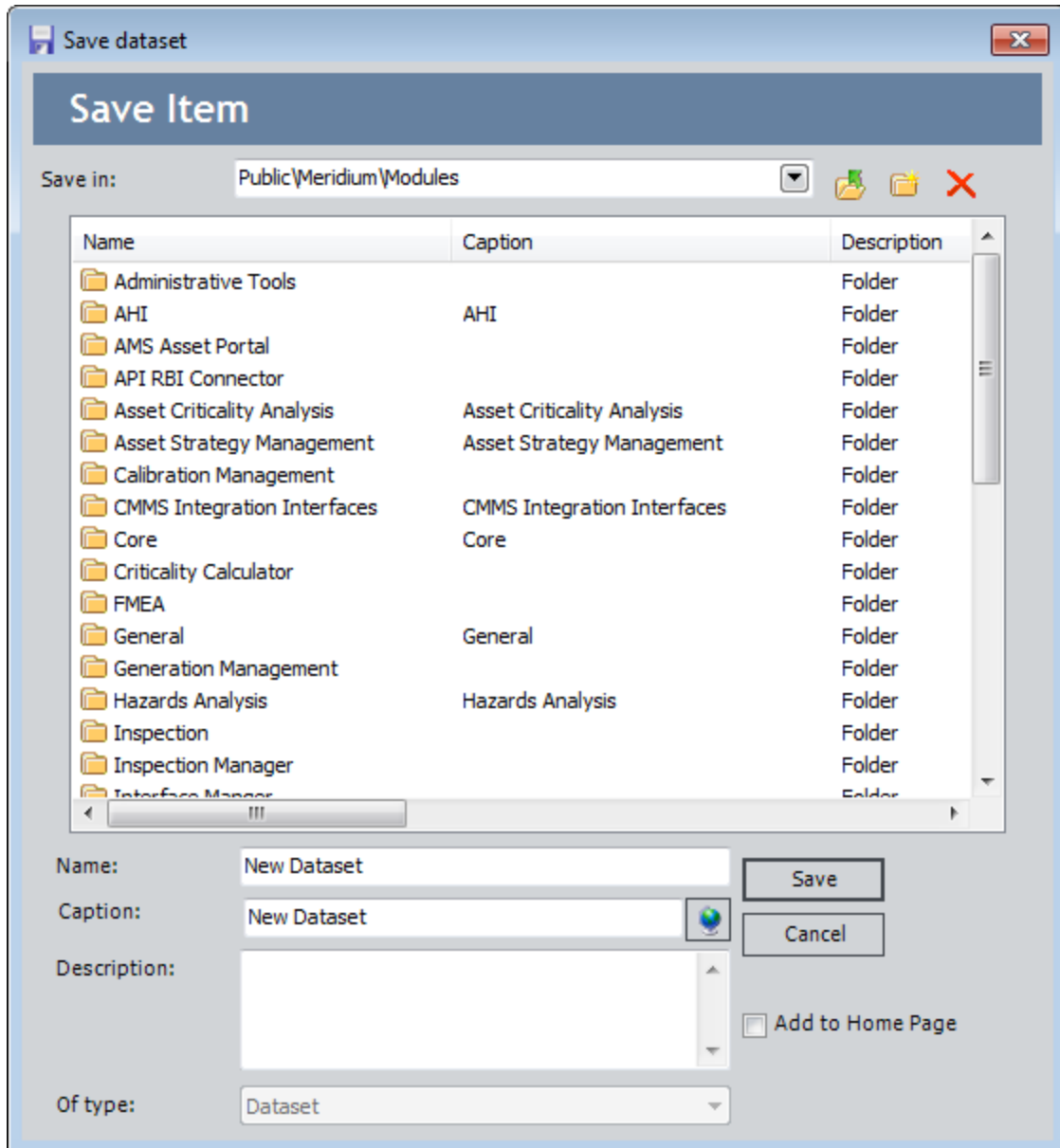
> **Note:** If you are viewing a saved dataset on the **Dataset Editor** page and you click the **Save As** link on the **Common Tasks** menu, the **Save dataset** dialog box will appear so that you can save a *copy* of the current dataset with a different name or to a different location. You can use the following instructions to complete that task, as the process is the same as saving a dataset for the first time.

2. Navigate to the Catalog folder in which you want to save the dataset.

3. In the **Name** text box, type a name for the dataset. The name is required and must be unique to the Catalog folder in which you are saving the dataset.

4. The **Caption** text box is populated automatically with the value that you typed in the **Name** text box.

5. If desired, click the 🌐 button to manage the translations for that string.

6. If desired, type a description of the dashboard in the **Description** text box. This is not required to save the dataset.

7. If desired, select the **Add to Home Page** check box if you want to add a link to the dataset to a Home Page. This is not required to save the dataset.

8. Click the **Save** button.

   If you:

   - Selected the **Add to Home Page** check box, the dataset is saved to the Catalog, and the Send to Home Page - Select Home Page dialog box is displayed.

     Depending upon your membership in the MI Power User Role Security Group and whether or not you are a Super User, you can use the **Send to Home Page - Select Home Page** dialog box provide a link to the dashboard to your Personal Home Page, a Group Home Page, or the Personal Home Page for a particular Security User.

   - Did not select the **Add to Home Page** check box, or you have completed the process of adding the dashboard to a Home Page, the **Save Dataset** dialog box closes, and the dataset is saved to the Catalog.

Deleting Datasets

These instructions provide details on deleting a saved dataset from the **Dataset Editor page**. You can also delete a saved dataset from the Catalog.

To delete a dataset:

1. In the Meridium APM Framework, open the dataset that you want to delete.

   The dataset appears on the **Dataset Editor** page.

2. On the **Common Tasks** menu, click the **Delete this dataset** link.

   A message appears, asking if you really want to delete this dataset.

3. Click the **Yes** button.

   The dataset is deleted from the Catalog.

   > **Note:** After the dataset has been deleted, the dataset column names remain displayed on the **Dataset Editor** page, but the dataset itself is no longer in the Catalog and cannot be modified or used in any way.

Creating Graphs

To create a new graph based on a query or dataset:

1. In the Meridium APM Framework, access the Graph Setup window.
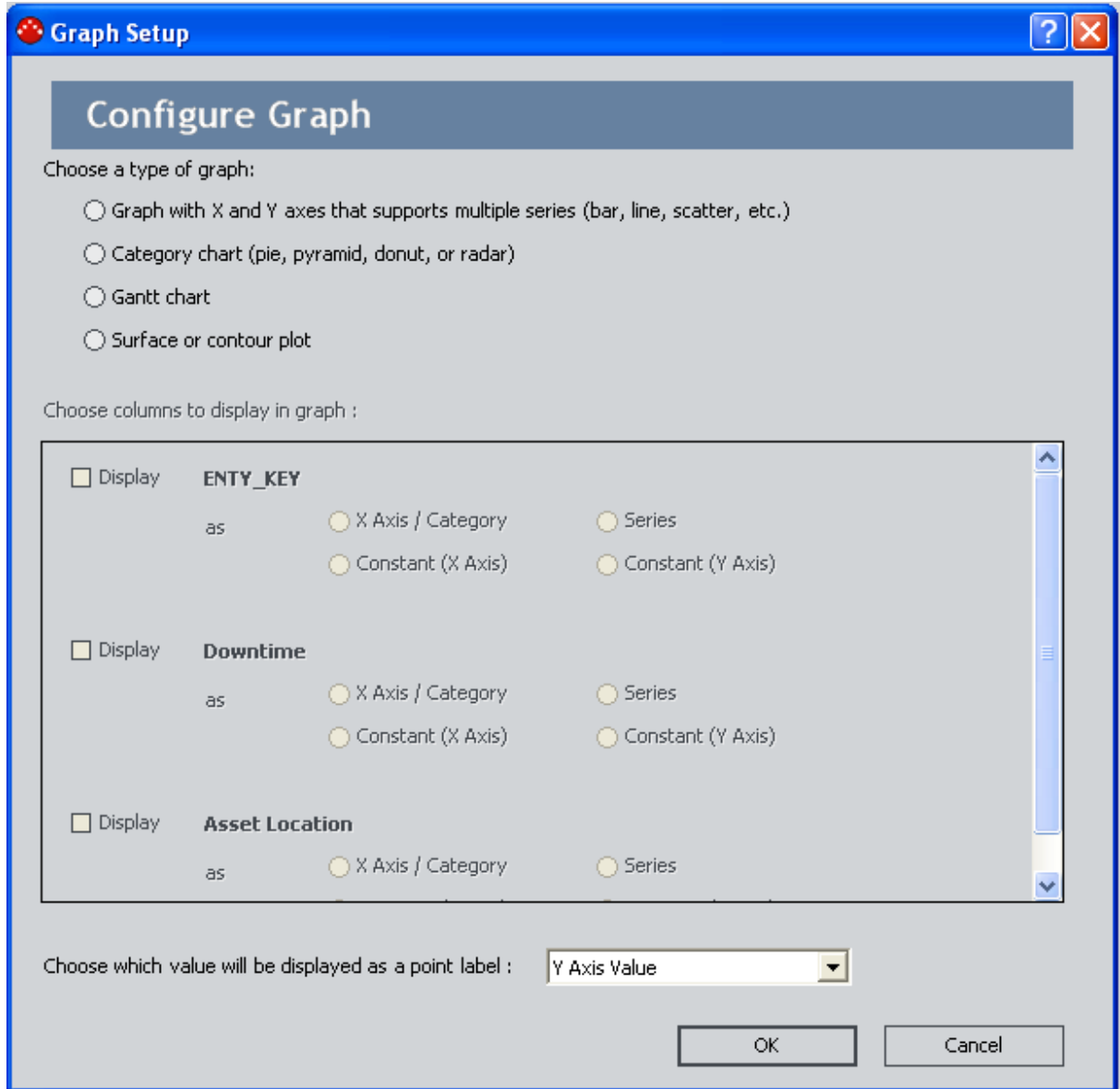
2. In the **Choose a type of graph** area, click the type of graph that you want to create.

3. In the **Choose columns to display in graph** area:

   - If the graph is based on a dataset, click the columns that you want to include in the graph.

   -or-

- If the graph is based on a query, click the fields that you want to include in the graph.

4. If desired, add hyperlinks to the graph. You can add hyperlinks only when the **Series** option is selected.

5. In the list in the **Choose which value will be displayed as a point label** area, click the value that you want to appear when you label a datapoint in the graph.

6. Click **OK**.

   The new graph is displayed on the **Graph Viewer** page.

   > **Note:**If you have made selections that do not fulfill the requirements of a basic graph, a message may appear, identifying the option that you need to select before you can view the graph.

Opening Graphs

Any graph that is stored in the Catalog can be opened on the **Graph Viewer** page.

To open an existing graph:

1. In the Meridium APM Framework, on the Meridium APM Framework toolbar, click **Graph**, and then click **Open Existing**.

   The **Open an existing Graph** dialog box appears.
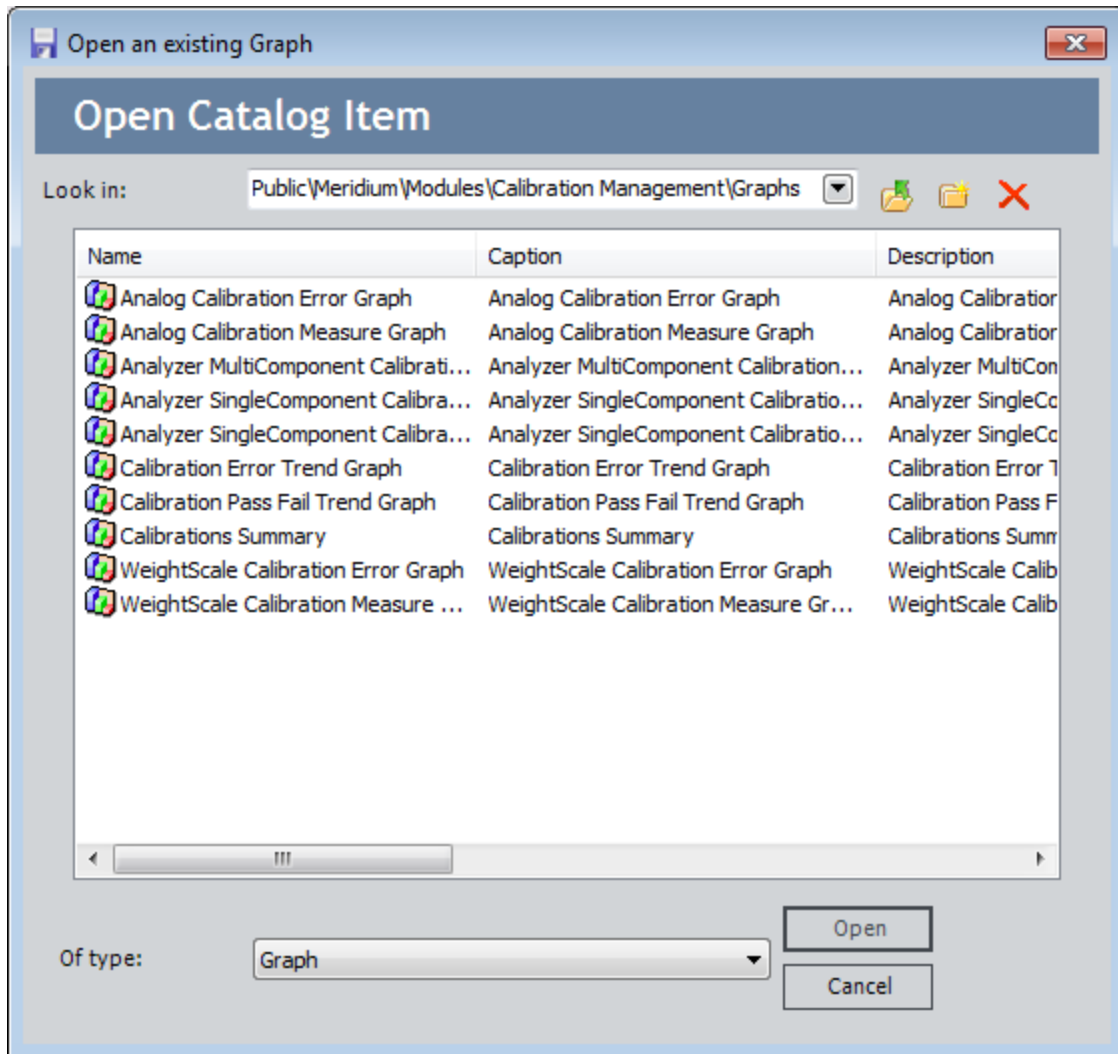
2. Navigate to the graph that you want to open, select it, and click the **Open** button.

   The graph appears on the **Graph Viewer** page.

Accessing the Graph Setup Window

You can access the **Graph Setup** window in any of the following ways:

- **From the Meridium APM Framework toolbar**. When you use this option, you will need to select from the Catalog the query or dataset from which you want to build a new graph.

  To access the Graph Setup window from the Meridium APM Framework toolbar:

  1. On the Meridium APM Framework toolbar, click **Graph**, and then click **Build New**.

     The **Choose a Query or Dataset for Graph** dialog box appears.

  2. Navigate to the query or dataset on which you want to base the graph, select it, and click the **Open** button.

     Note the following:

     - The value in the **Of type** list is always set to *Query* by default. To see the datasets in a Catalog folder, you will need to select *All* or *datasets* in the **Of type** list.

     - Only Select or Crosstab queries can be used to create a graph.
       The **Graph Setup** window appears.

- **From query results**. This option allows you to build a new graph based on the current query.

  To access the Graph Setup window from query results:

  1. Open or create a query, and view the results.

  2. On the **Other Tasks** menu, click the **Create a Graph** link.

     The **Graph Setup** window appears.

     > Note: The **Create a Graph** link is disabled until you save the graph.

- **From the Dataset Editor page**. This option allows you to build a new graph based on the current dataset.

  To access the Graph Setup window from the Dataset Editor page:

  1. Open or create a dataset and view it on the **Dataset Editor** page.

  2. On the **Other Tasks** menu, click the **Create a new graph** link.

     The **Graph Setup** window appears.

     > Note: The **Create a new graph** link is disabled until you save the dataset.

- **From the Graph Viewer page**. This option allows you to modify the graph that is currently displayed.

  To access the Graph Setup window from the Graph Viewer page:

  1. On the **Graph Viewer** page, on the **Graph Tasks** menu, click the **Alter Configuration** link.

A confirmation message appears, indicating that changing the graph configuration may cause some graph settings to be lost.

2. Click the **Yes** button to continue.

   The **Graph Setup** window appears, displaying the graph's current settings.

The following image shows an example of the **Graph Setup** window that was accessed by opening a query.
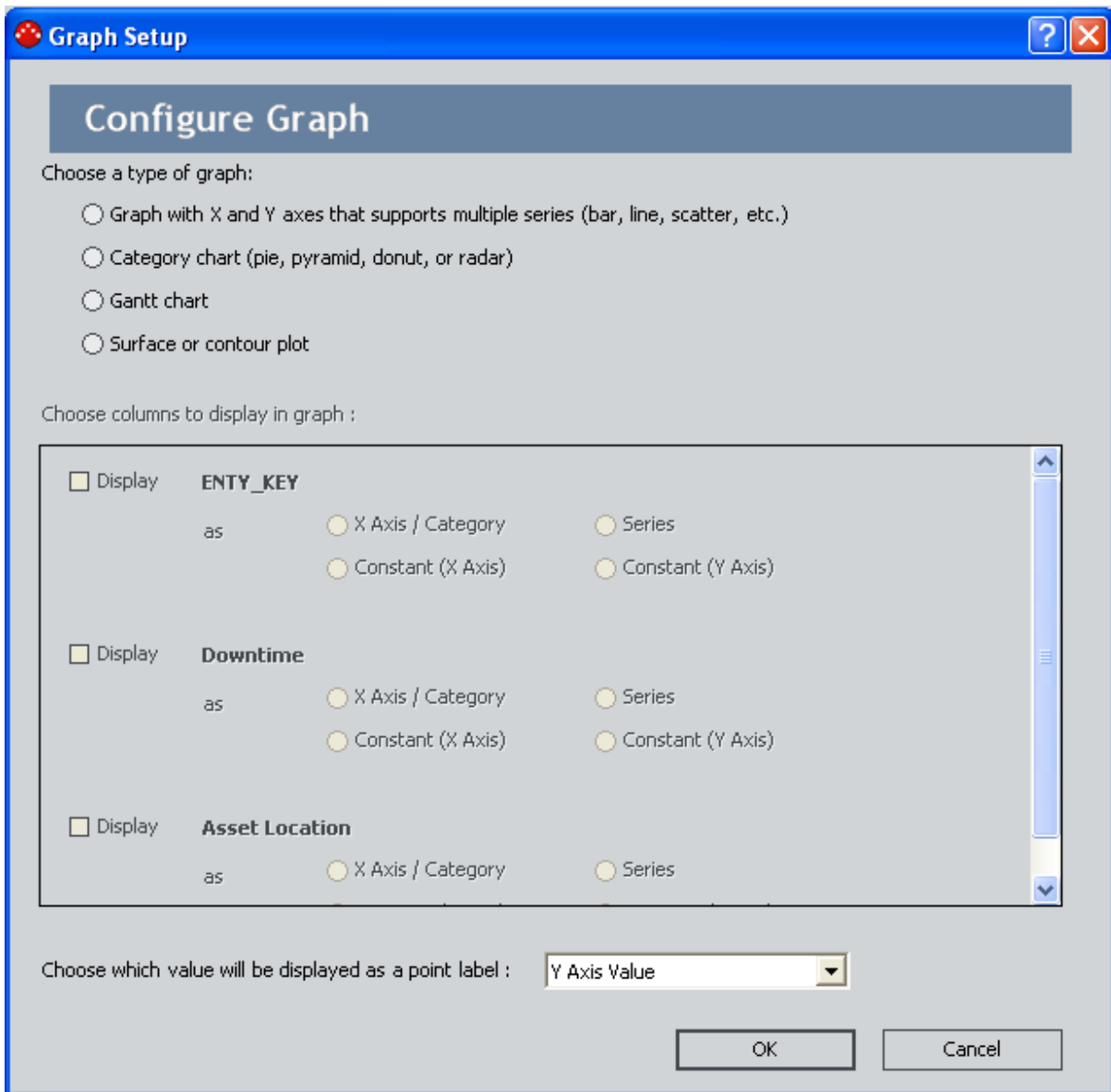
Aspects of the Graph Setup Window
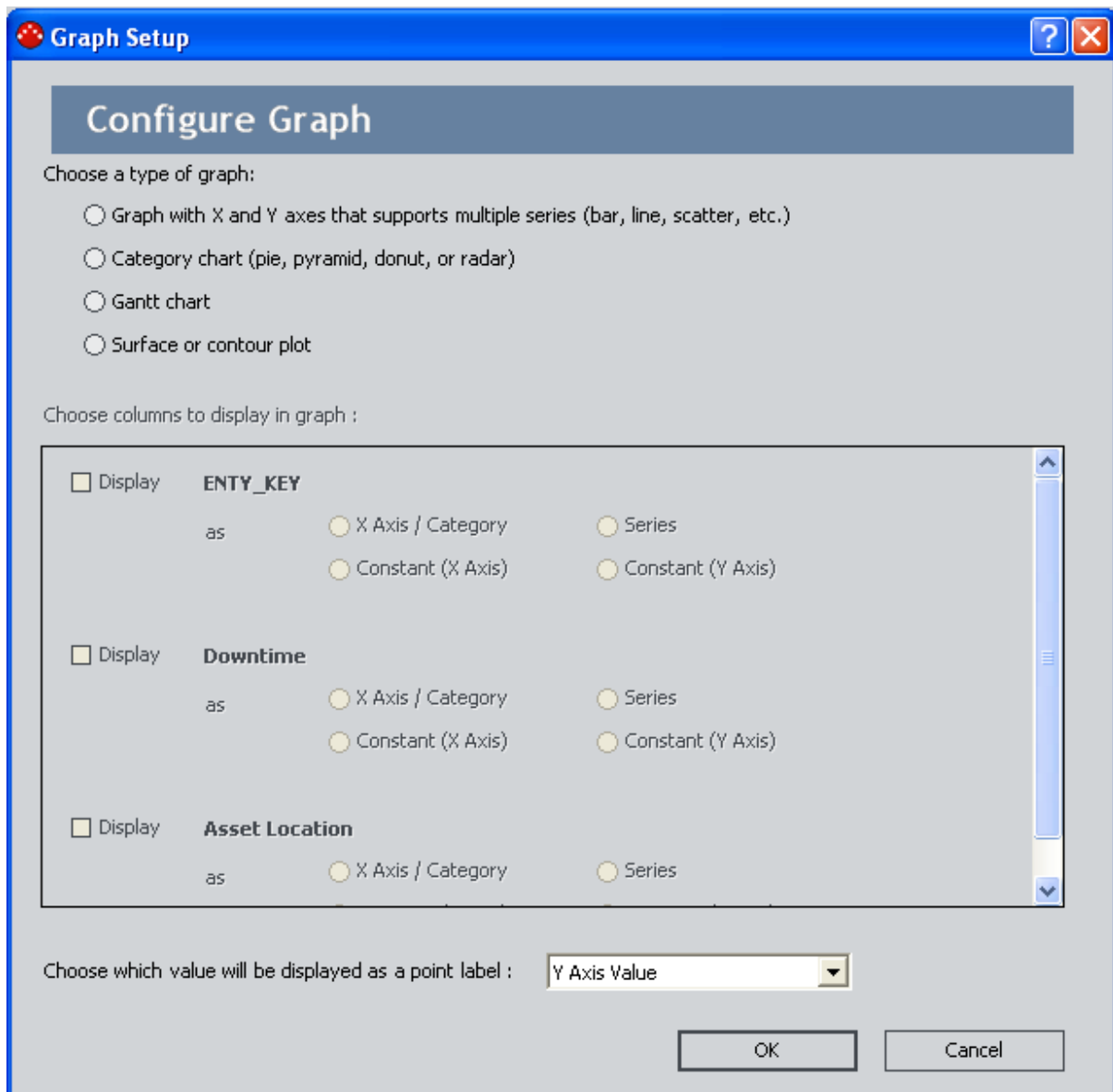
The Graph Setup window contains three areas:

- **Choose a type of graph:** Displays options corresponding to the four main types of graphs that you can create:
  - A graph with an x-axis and a y-axis.
  - A category graph.
  - A gantt graph.
  - A surface or contour graph.

- **Choose a type of graph:** Displays options corresponding to the four main types of graphs that you can create:
    - A graph with an x-axis and a y-axis.

    - A category graph.

    - A gantt graph.

    - A surface contour graph.
- **Choose columns to display in graph:**
    - If the graph is based on a dataset, this area displays the column headers of the columns that are displayed in the dataset.

    - If the graph is based on a query, this area displays the field captions of the field that are included in the query.

    > **Note:** In a graph, you can display values that are stored in *any* field that is included in the underlying query, even if it is not actually displayed in the query results.

    After you select the columns or fields whose values you want to display in the graph, you will need to select additional options to indicate how those values should be displayed (e.g., as bars on a bar graph or as lines on a line graph).

    Note that some options will be disabled based on the type of data that is stored in the selected column or field and the other graph selections that you have made. Other options will not appear until another option is selected.

    For example, after you have selected a column as a series, additional options will appear that let you select the type of graph from the **Gallery** list.

- **Choose which value will be displayed as a point label:**
    - If the graph is based on a dataset, this area displays a list that contains the column headers of the columns that are displayed in the dataset, and the default selection, **Y Axis Value**.

    - If the graph is based on a query, this area displays a list that contains the field captions of the fields that are included in the query, and the default selection **Y Axis Value**.

    The value that is selected in this list indicates the value that will be displayed when you [show the value of a datapoint in the graph](). For example, if you select *Asset ID*, when you show the value of a datapoint in the graph, the displayed value will be the Asset ID of the selected datapoint.

Selecting a Graph Type

To select the type of graph that you want to create:

1. In the Meridium APM Framework, on the **Graph Setup window**, in the **Choose a type of graph** area, select the option that corresponds to the type of graph that you want to create. You can select one of the following options:

   - Graph with x and y axes that support multiple series (bar, line, scatter, etc.)
   - Category chart (pie, pyramid, doughnut, or radar)
   - Gantt chart
   - Surface or contour plot

2. Depending on the type of graph that you selected, you must also select certain fields to include in the graph.

3. When you are finished making all your selections, click **OK**.

   The graph appears on the **Graph Viewer** page.

About Selecting Columns or Fields to Include in a Graph

The **Choose columns to display in graph** area of the **Graph Setup** window displays a list of the columns or fields that exist in the underlying dataset or query.

The columns or fields that you choose to include will depend on the data that you want to display in the graph. For example, consider a Crosstab query that queries the Motor family and returns for each manufacturer counts of active and inactive motors for each manufacturer. The query results might look like this:

Manufacturer	Active	Inactive
ACME	500	10
American Motors	450	3
Best Electric	1000	6
Master Machinery	623	19

A graph that you build from this query could display a visual representation of the *inactive* motors for each manufacturer:
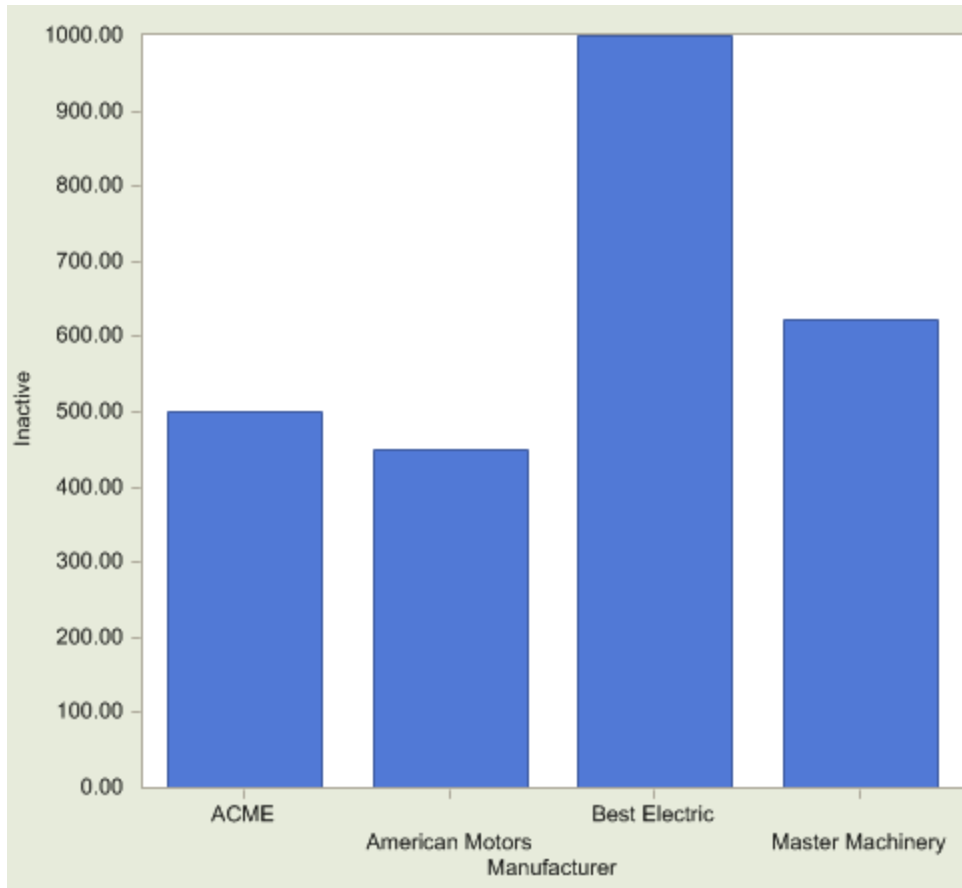
...the *active* motors for each manufacturer:

...or *both* active and inactive motors for each manufacturer:

You might want to create multiple graphs from a single query to show as many visual representations of the data as possible.

If you are creating a graph with an x- and y-axis, as you choose which columns or fields to include in the graph, keep in mind that the y-axis displays the value that is plotted (e.g., the number of active or inactive motors), and the x-axis displays the label or category for the plotted points (e.g., the manufacturer).

In a graph that does not have an axis (i.e., a pie or doughnut), instead of axes, you will see a series of values within categories (i.e., the bars without the chart).

Selecting Columns or Fields to Include in a Graph

To select columns or fields to include in a graph:

1. In the Meridium APM Framework, on the **Graph Setup window**, in the **Choose columns to display in graph** area, select the check boxes next to the columns or fields that you want to include in the graph.

2. For each column or field that you selected, select one of the following options which will be enabled depending on your selections:

   - **X Axis/Category:** If the graph contains an x-axis or y-axis, the data in the column or field will be plotted on the graph as the x-axis. If, however, your graph does not contain an x-axis or y-axis, the data in the column or field will appear as the category on the graph (e.g., a slice in a pie graph).

   - **Series:** The column or field that represents the variable data that will be plotted on the graph. After you select **Series** as the plot type for a column or field:

     - The **Left** and **Right** y-axis options become available. You will need to select the option that corresponds to the side of the graph on which you want the y-axis to be displayed.

     - The **Gallery** list becomes available. In the **Gallery** list, you can select the desired graph type. The types that are available depend upon the main graph type that you selected in the **Choose a type of graph** area.

     - The **Include Hyperlink option** becomes enabled. If you include a hyperlink allows you you to click on a point on the graph to open a related page.

   - **Constant (X Axis)** The column or field that will be plotted on the x-axis as a constant.

   - **Constant (Y Axis)** The column or field that will be plotted on the y-axis.

     > **Note:** For the y-axis, you will need to choose whether you want the scale to appear on the right or the left by selecting the appropriate option button.

   This constant option is useful if you want to compare a series of values to a single unchanging value. If you select a column or a field to be plotted as the constant, the Meridium APM system will plot a single value along the specified axis to create a line across the entire graph, where the line serves as an indicator of how close each value in the series is to the constant value. A constant y-axis value will create a horizontal line. A constant x-axis value will create a vertical line.

3. When you are finished making all your selections, click **OK**.

   The graph appears on the **Graph Viewer** page.

The Right and Left Y-Axis Scales

When you choose a column or field to be plotted on the y-axis, you must specify the side of the graph on which the axis will appear: left or right. You can do so using the options that are available on the **Graph Setup** window.

If you choose to plot more than one column or field on the y-axis and the values in those columns or fields vary greatly, using a right and left y-axis can improve the usability of the graph.

Consider our example from before, where we built a graph using counts of active and inactive motors for a set of manufacturers. In this example, if we build a graph to display *both* active and inactive counts using only a *left* scale, the inactive counts are difficult to see. This is because the graph scale is based on the number of *active* motors, which are much higher than the number of inactive motors, as shown in the following image.

If, instead, we plot the *active* motors on the *left* and the *inactive* motors on the *right*, the bars for the inactive counts are easier to see. This is because the size of the bars has been adjusted according to a scale that is more appropriate to the size of the numbers, as shown in the following image.

About Adding Hyperlinks to a Graph

You can add hyperlinks to a graph to access any URL-addressable Meridium APM module or any external website. When a hyperlink has been added to a graph, clicking a data-point on the graph will take you to the location defined for the link.

> **Note:** If you add a hyperlink to a pie, doughnut, or pyramid graph, the URL will not be invoked when you click the graph.

When you add a hyperlink to a graph, you can configure it such that when you click a point on the graph (i.e., clicked point), a specific value from the graph will be passed in as a parameter value to the URL. You can pass any of the following values into the URL:

- The y-axis value of the clicked point.
- The x-axis value of the clicked point.
- The value stored in a specific field in the record associated with the clicked point.
- The Family Key of the record associated with the clicked point.
- The Entity Key of the record associated with the clicked point.
- All the values for any previously accessed graph.

> **Note:** For details on passing a parameter value to a URL, see the URLs section of the Meridium APM Help.

Adding a Hyperlink to a Graph

The following instructions provide general information for adding a hyperlink to a graph.

To add a hyperlink to a graph:

1. In the Meridium APM Framework, access the Graph Setup window, locate the column or field designated as the **Series**, and select the **Include Hyperlink** check box.

   The **Include Hyperlink** text box becomes enabled.
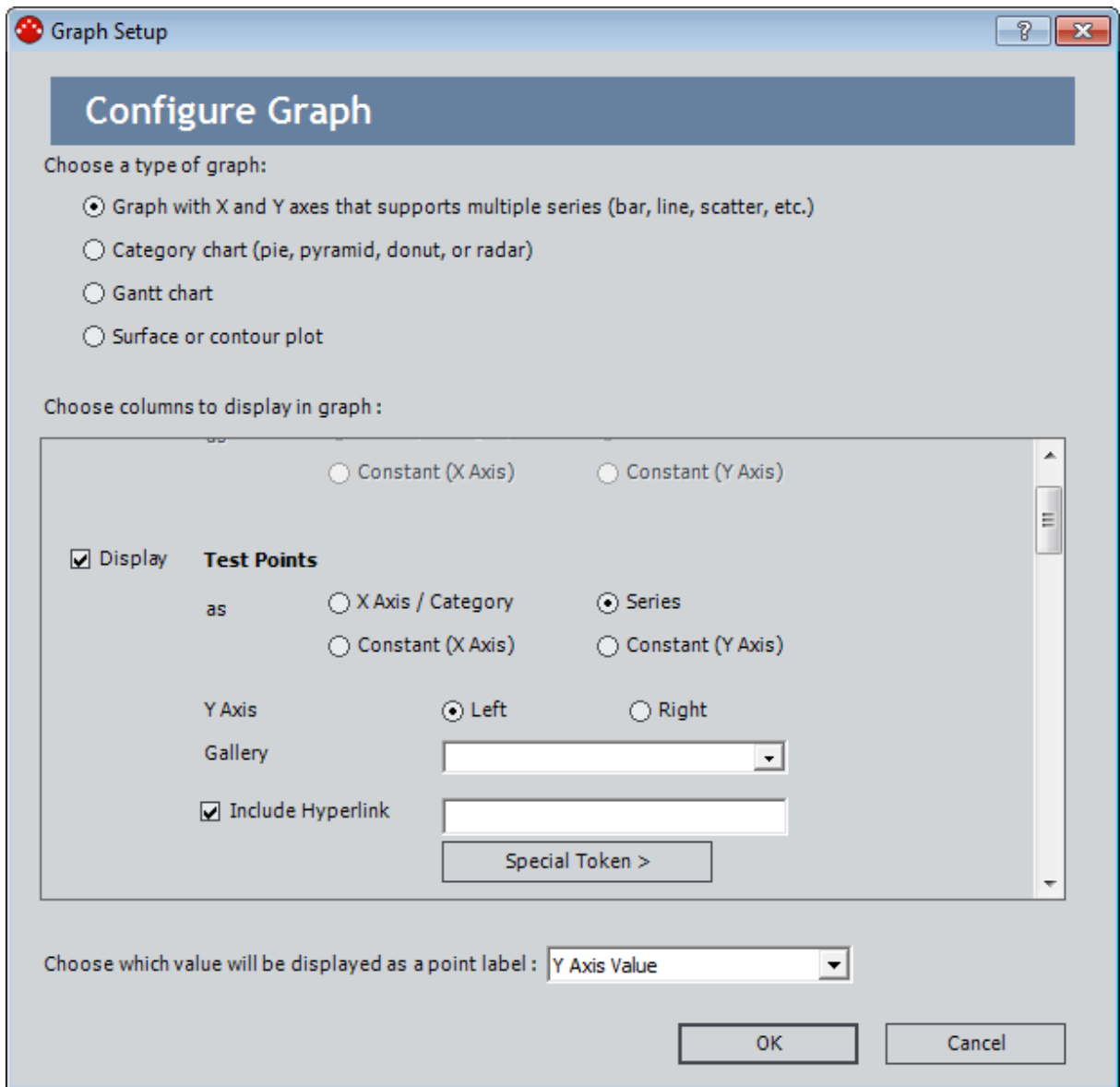
2. In the **Include Hyperlink** text box, type the desired hyperlink, including the protocol portion of the URL. You can type either an internal URL (e.g., meridium://Inspection) or an external URL (e.g., http://www.meridium.com).

3. If you want to pass values in for a parameter in your URL, position your cursor after the equal sign for that parameter, click the **Special Token** button, and then click:

   - **Y Axis Value for Clicked Point:** To pass in the y-axis value of the clicked point as the value for the URL parameter.

   - **X Axis Value for Clicked Point:** To pass in the x-axis value of the clicked point as the value for the URL parameter.

   - **Available Column Values for Clicked Point:** To pass in the value stored in a specific field or column in the underlying query or dataset as the value for the URL parameter. After you click **Available Column Values for Clicked Point**, you will see a submenu that displays all fields or columns that are included in the underlying query or dataset. Select a field to pass in the value that is stored in that field for the datapoint that you click on the graph.

   - **Family Key for Clicked Point:** To pass in the Family Key of the clicked point as the value for the URL parameter.

   - **Entity Key for Clicked Point:** To pass in the Entity Key of the clicked point as the value for the URL parameter.

   - **Graph URL Query Parameters:** To pass in all the values for any previously accessed graph.

   > **Note:** Selecting an option inserts the text **[YValue]**, **[XValue]**, **[QueryString]**, **[FamilyKey]**, **[EntityKey]**, or **[Field Name]** (where **Field Name** is the caption of the field whose value you are passing to the parameter) wherever your cursor is positioned. If you prefer, you can type this text manually. Additionally, you can insert it multiple times as needed for the URL that you are constructing.

4. Click **OK** to close the **Graph Setup** window.

   The graph returns to focus. You can click any datapoint on the graph to view the destination of the hyperlink that you added.

Linking from a Graph to Another Meridium APM Feature

You can add to a graph [hyperlinks](#) that will provide users access to an internal or external location. One unique feature of hyperlinks in graphs is that for any URL that accepts parameters, you can pass in as the parameter value the x-axis or y-axis value associated with the clicked point.

For example, consider the following graph that shows the total cost of failures for motors in the database.
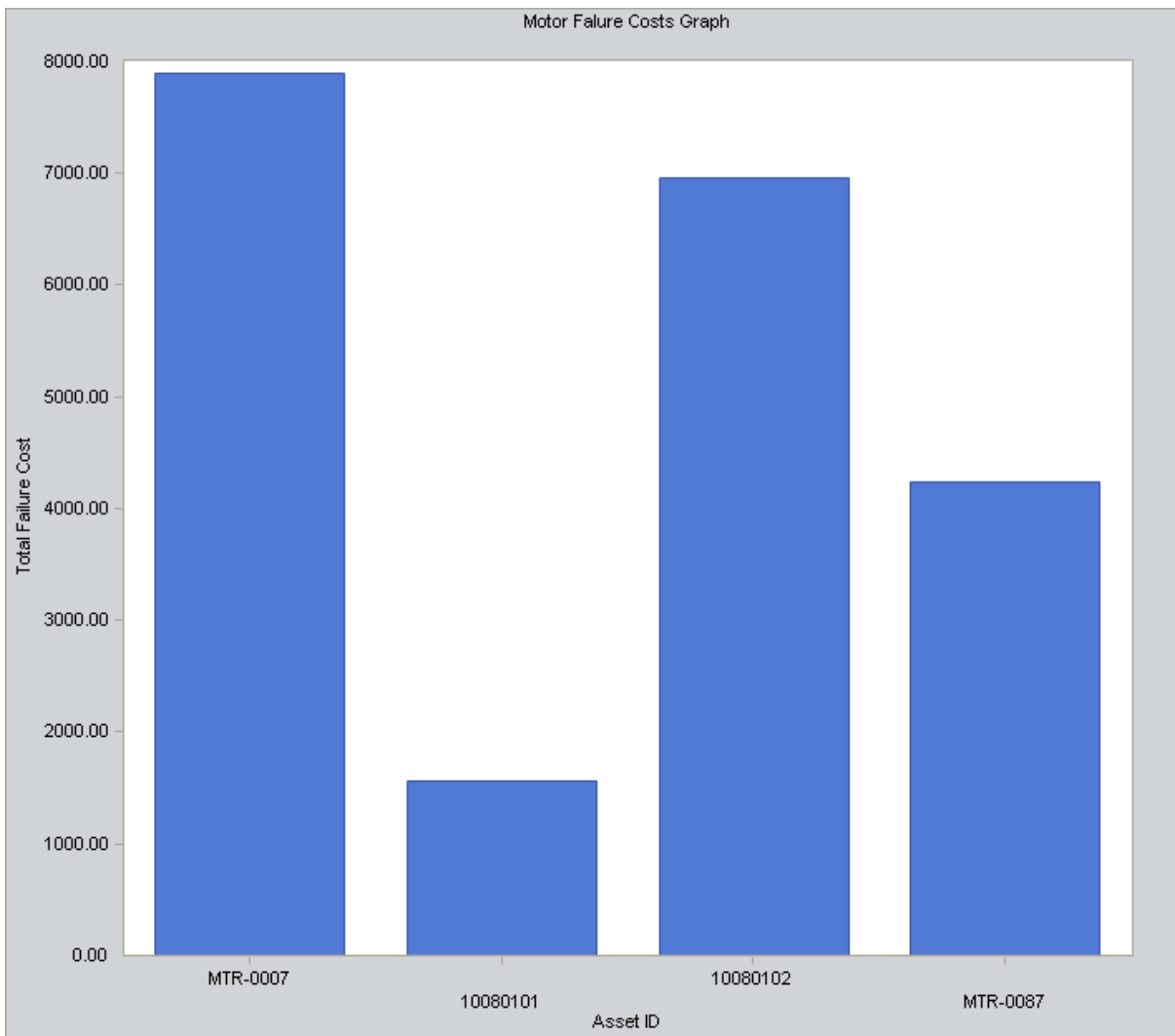
You may want to add a hyperlink to the graph that would provide the user with additional information about the Motor records that are displayed on the x-axis. To do this, you could construct a query that returns the desired information, such as a list of Inspection records that are linked to the Motor records. You would add to the query a prompt that accepts an Asset ID and returns a list of the related Inspection records, which, in

turn could be linked to the records in Record Manager. The URL that you add to the graph might look something like this:

meridium://Registry/QueryBuilder?QueryPath=Public\Meridium\Queries\Motor Inspections&p0=[XValue]

...where:

- Path=Public\Meridium\Queries\Motor Inspections specifies that you want to execute the Motor Inspections query, which is stored in the Catalog folder \\Public\Meridium\Queries.

- p0=[XValue] specifies that for the first prompt, you want to pass in the x-axis value from the point on the graph that is clicked, which in this case, will be the Asset ID of a motor. Note that specifying *p0=[YValue]* would pass the y-axis value from the clicked point in for the prompt.

> Note: You can also pass in as parameter values any value that is not actually displayed on the graph but is included in the underlying query or dataset.

Linking from One Graph to Another

The previous topic provided a simple example of linking from a graph to another Meridium APM feature, specifically, a query. Here, we extend this concept by explaining how you might link from one graph to another graph that provides more detailed information about the first graph. Consider an example where:

- The first graph displays the total cost of failures for the Motor records in your data-base.
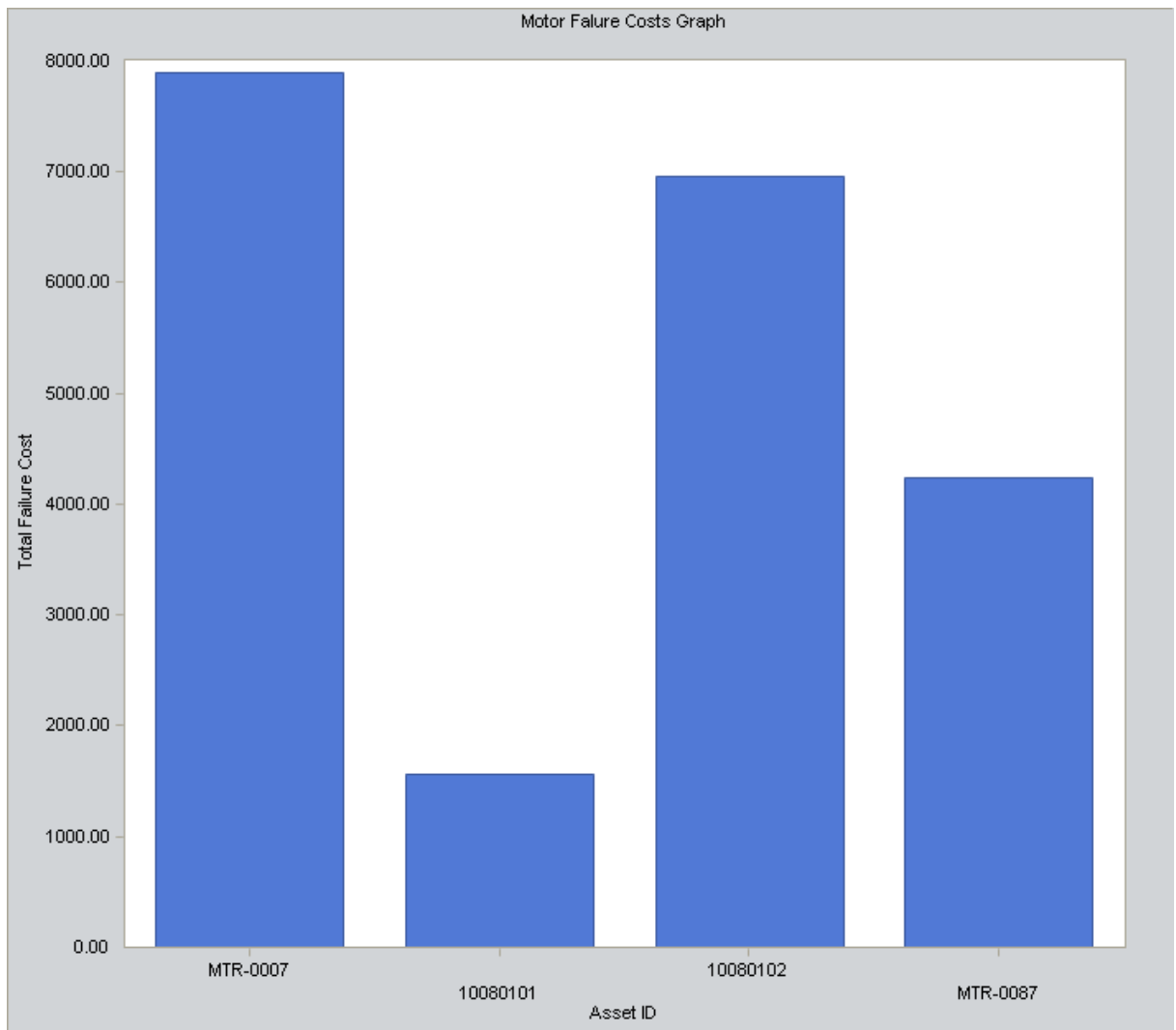
- The second graph displays for a given Motor record a breakdown of the total fail-ure cost by category, such as labor cost and material cost.
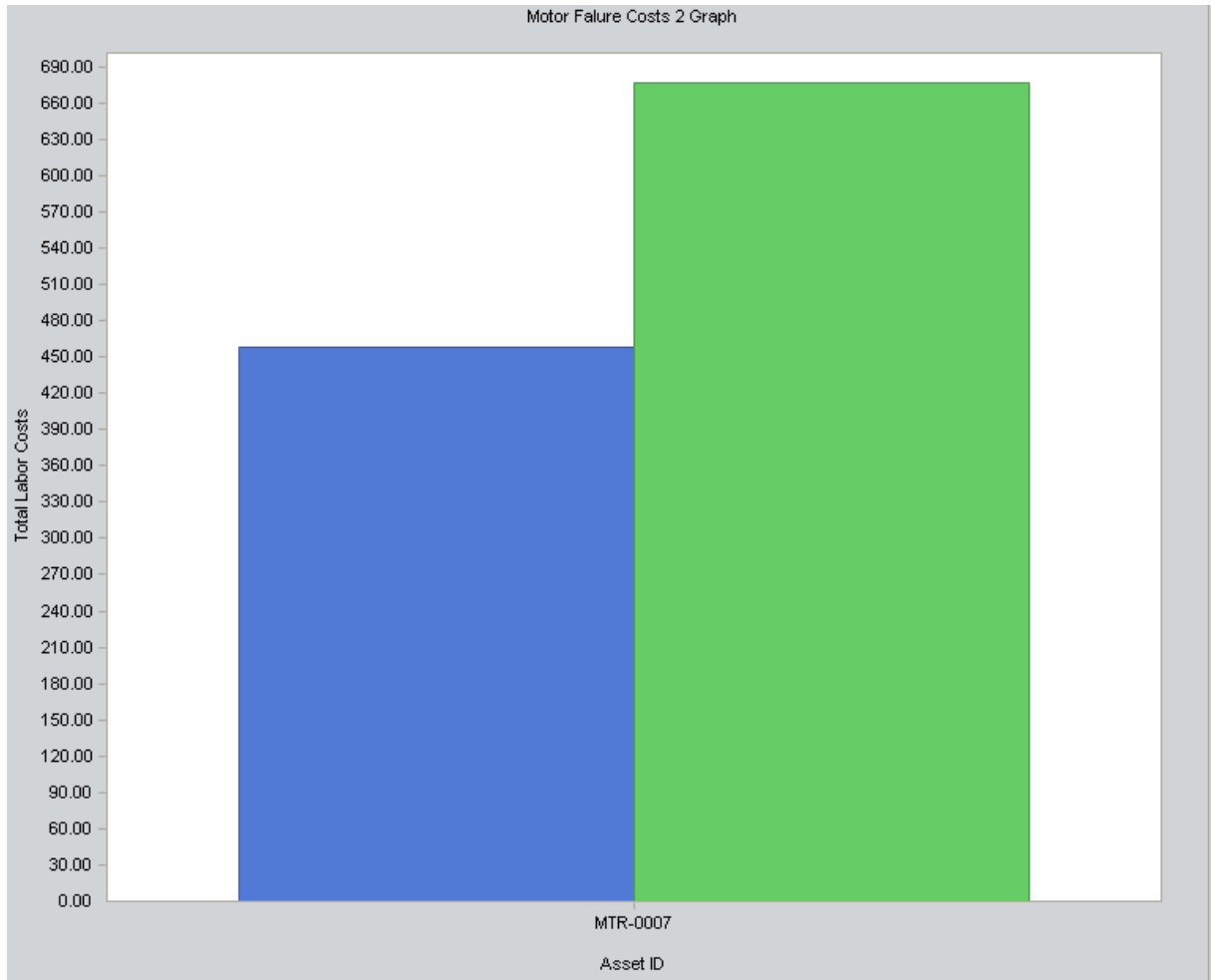
After you have created the two graphs, you can add a hyperlink to the first graph so that users will be able to access the more detailed graph from the first graph. The URL might look something like this:

meridium://Registry/GraphViewer?GraphPath=Public\Meridium\Graphs\Failure Breakdown&p0=[XValue]

...where:

- **GraphPath=Public\Meridium\Graphs\Failure Breakdown** specifies that you want to open the Failure Breakdown graph, which is stored in the Catalog folder \\Public\Meridium\Graphs.

- **p0=[XValue]** specifies to pass the value from the x-axis of the clicked point to the prompt in the query associated with the second graph. In this case, the second graph would need to be based on a query that contains a prompt that accepts an Asset ID and then returns for that piece of equipment or location the detailed failure cost information.

You could further extend this example by linking from the second graph to a third graph that shows even more detail. Note that as you pass values from one graph to the

next, the system saves them temporarily in memory. Therefore, in any hyperlink that you add to a graph you can use the [QueryString] option to pass in to the URL ALL the values that have been passed from one graph to the next, up to that point. For example, if you pass the Asset ID from Graph A to Graph B and the Failure Cost from Graph B to Graph C, if you add a hyperlink to Graph C, you could use the [QueryString] option to pass the Asset ID and Failure Cost to Graph D.

Example: Passing In the Y-Axis Value

The following example explains how to create a hyperlink such that when you click a datapoint on the graph, a new record will be opened in the Record Manager and the Design Pressure (Tubeside) field will be populated automatically with the y-axis value of the record that is associated with the datapoint that you clicked on the graph.

This example assumes that the graph is based on a query that returns the following results.

Asset ID	Design Pressure (Shellside)	Design Pressure (Tubeside)
HE-1608	200	475
HE-1609	200	475
HE-1612	215	475
HE-1613	215	475
HE-360	50	100
HE-2445	215	296
HE-361	50	100
HE-379	50	150
HE-380	50	150

Note that the Asset ID, Design Pressure (Shellside), and Design Pressure (Tubeside) fields are included in the query.

After the graph has been built from the query, it would look like the following image, assuming that you plot the Asset ID along the x-axis, the Design Pressure (Shellside) along the left y-axis, and the Design Pressure (Tubeside) along the right y-axis.
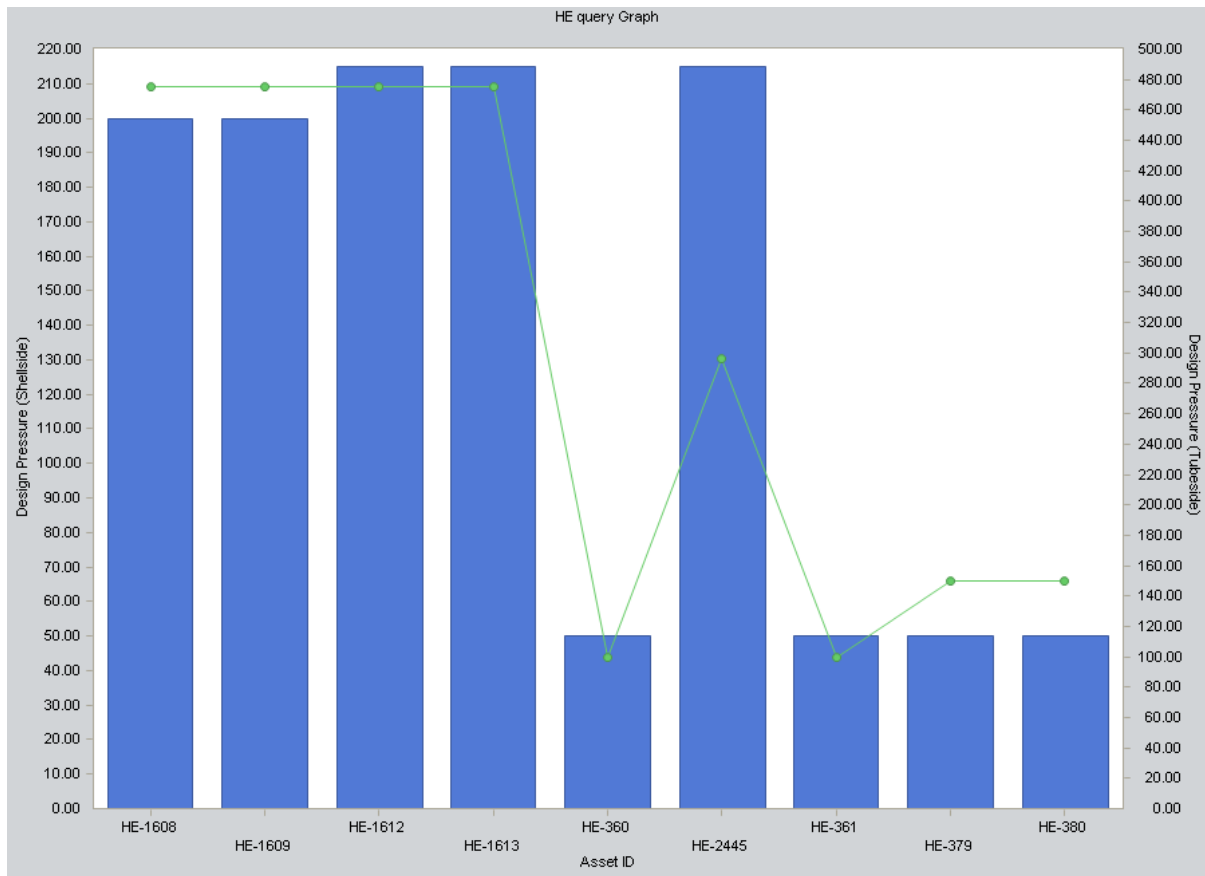
Suppose all of your equipment or locations in this graph belong to the Shell and Tube Heat Exchanger family. You might want to create a new Shell and Tube Heat Exchanger record from this graph and populate the Design Pressure (Tubeside) field automatically with the same value that exists in the Design Pressure (Tubeside) field in the record whose datapoint you click on the graph. You can create a hyperlink to do so.

To create this hyperlink, on the **Graph Setup** window, for the field that you are displaying as the series, you would need to add a URL that will open a new record in the Record Manager. The URL would look like this:

meridium://Registry/RecordManager?EntityKey=0&FamilyKey=[FamilyKey]

...where:

- The **EntityKey** parameter value is **0**, which creates a new record.

  -and-

- The **FamilyKey** parameter value is **[FamilyKey]**, which will create a record in the family of the record whose datapoint you click on the graph.

> Œ**Hint:** You can let Meridium APM automatically insert the value *[FamilyKey] into your URL* by selecting **Family Key for Clicked Point** on the **Special Token** submenu.

So far, the URL will allow you to click a datapoint on the graph and create a new record in the same family as that of the record that is associated with the clicked datapoint. If you want to populate a field in the new record automatically with the y-axis value in the record associated with the datapoint that you click, however, you will need to add another parameter to the URL: [YValue].

> Œ Hint: Because the y-axis displays Design Pressure (Tubeside) values, you can use the [YValue] parameter in the URL. If the y-axis did not display Design Pressure (Tubeside) values and you still wanted to populate that field in the new record, you would need to use the [FieldID] parameter.

To add the [YValue] parameter to the URL, after the last parameter value, which in this case is [FamilyKey], you would need to type the field ID of the field that you want to populate, followed by the equal sign (=). Then, click the **Special Token** button, and then click **Y Axis Value for Clicked Point** on the submenu, as shown in the following image.
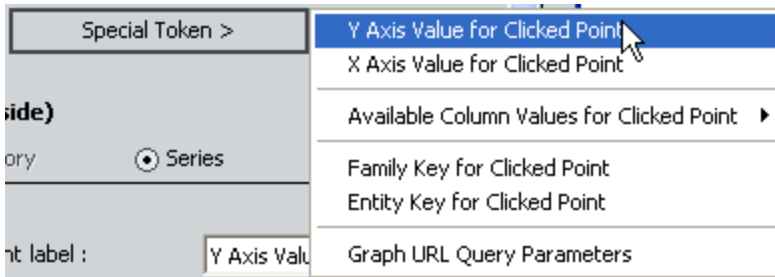
This step will automatically insert the text *[YValue]* into the URL. So, the complete URL would look like this:

meridium://Registry/RecordManager?EntityKey=0&FamilyKey=[FamilyKey]&AFHST_ DESPTB_NBR=[YValue]

When you click a datapoint on the graph:

- The URL will be executed.
- A new Shell and Tube Heat Exchanger record will appear in the Record Manager.
- The Design Pressure (Tubeside) field will be populated automatically with the y-axis value of the record whose datapoint you clicked on the graph.

Example: Passing In the Y-Axis Value

	Shell Side	Tube Side
Operating Pressure		
Operating Temperature		
Design Pressure		300 (Pounds/Sq Inch Gage)
Design Temperature		
Material Handled		
Hydrotest Pressure		
Character of Fluid		
Number of Passes		
MAWP Corroded		
Max. Temp at MAWP Corroded		
Latent Heat		
Min. Fouling Resistance		
Des.Press. @ Degrees F		

Example: Passing In the X-Axis Value

The following example explains how to create a hyperlink such that when you click a datapoint on the graph, a new record will be opened in the Record Manager, and the Asset Manufacturer field will be populated automatically with the x-axis value of the record whose datapoint you clicked on the graph.

This example assumes that the graph is based on a query that returns the results shown in the following image.

	Asset Manufacturer	Tank Height
▶	CBI	25
	IMC FAB	28
	PACIFIC	100
	ATLANTIC	130
	HYDROCARBO	120

Note that the Asset Manufacturer and Tank Height fields are included in the query.

After the graph has been built from the query, it would look like the following image, assuming that you plot the Asset Manufacturer along the x-axis and the Tank Height along the y-axis.

Suppose all of your equipment represented by the graph belong to the Tank family. You might want to create a new Tank record from this graph and populate the Asset Manufacturer field automatically with the x-axis value of the record whose datapoint you click on the graph. You can create a hyperlink to do so.

To create this hyperlink, on the **Graph Setup** window, for the field you are displaying as the series, you would need to add a URL that will open a new record in the Record Manager. This URL would look like this:

meridium://Registry/RecordManager?EntityKey=0&FamilyKey=[FamilyKey]

...where:

- The **EntityKey** parameter value is **0**, which creates a new record.

  -and-

- The **FamilyKey** parameter value is **[FamilyKey]**, which will create a record in the family of the record that is associated with the datapoint that you click on the graph.

  > ŒHint: You can let Meridium APM automatically insert the value *[FamilyKey]* into your URL by selecting **Family Key for Clicked Point** on the **Special Token** submenu.

So far, the URL will allow you to click a datapoint on the graph and create a new record in the same family as that of the record that is associated with the datapoint that you click. If you want to populate a field in the new record automatically with the x-axis value of the record of the datapoint you click, you will need to add another parameter to the URL: [XValue].

> ŒHint: Because the x-axis displays Asset Manufacturer values, you can use the [XValue] parameter in the URL. If the x-axis did not display Asset Manufacturer values and you still wanted to populate that field in the new record, you would need to use the [FieldID] parameter.

To add the [XValue] parameter to the URL, after the last parameter value, which in this case is [FamilyKey], you would need to type the field ID of the field that you want to populate, followed by the equal sign (=). Then, click the **Special Token** button, and then click **X Axis Value for Clicked Point** on the submenu, as shown in the following image.
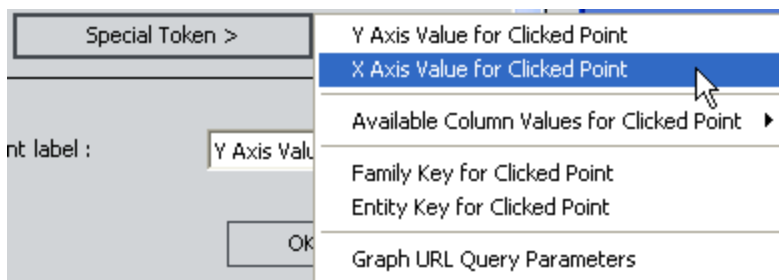
This step will automatically insert the text *[XValue]* into the URL. So, the complete URL would look like this:

meridium://Registry/RecordManager?EntityKey=0&FamilyKey=[FamilyKey]&ASSET_MANUF_CHR=[XValue]

When you click a datapoint on the graph:

- The URL will be executed.
- A new Tank record will appear in the Record Manager.
- The Asset Manufacturer field will be populated automatically with the x-axis value of the record whose datapoint you clicked on the graph.

	Value(s)
▶ Asset ID	
Asset Description	
Asset Type	
Asset Manufacturer	HYDROCARBO
Asset Model Number	
Asset Serial Number	
Asset Installation Date	
Asset Status	
Critical Asset ?	☐
Spared ?	☐
Asset Service	
Asset P&ID Drawing Number	
Shutdown Req. for Maintenance?	☐
Asset Additional Information	

Example: Passing in the Family Key

The following example explains how to create a hyperlink such that when you click a datapoint on the graph, a new record will be opened in the Record Manager in the same family as the asset whose datapoint you clicked on the graph.

Suppose all of your assets in a graph belong to the Tank family. You might want to create a new Tank record from this graph. To do so, you can create a hyperlink so you can click any datapoint to create a new Tank record in the Record Manager.

To create this hyperlink, on the **Graph Setup** window, for the field that you are displaying as the series, you would need to add a URL that will open a new record in the Record Manager. This URL would look like this:

meridium://Registry/RecordManager?EntityKey=0

...where the **EntityKey** parameter value is **0**, which creates a new record.

To indicate that when you click a datapoint on the graph you want to create a new record in the family of the record that is associated with the clicked datapoint, you need to add the FamilyKey parameter to the URL.

To add the [FamilyKey] parameter value to the URL, after the Entity Key parameter value, which in this case is zero (0), you would need to type the FamilyKey parameter, followed by the equal sign (=). Then, click the **Special Token** button, and then click **Family Key for Clicked Point**.

This step will automatically insert the text *[FamilyKey]* into the URL.

The complete URL would look like this:

meridium://Registry/RecordManager?EntityKey=0&FamilyKey=[FamilyKey]

When you click a datapoint on the graph:

- The URL will be executed.
- A new record will appear in the Record Manager in the family of the clicked datapoint.

Example: Passing in the Entity Key

The following example explains how to create a hyperlink such that when you click a datapoint on the graph, the record that is associated with the datapoint that you clicked will be opened in the Record Manager.

To create this hyperlink, on the **Graph Setup** window, for the field that you are displaying as the series, you would need to add a URL that will open the Record Manager. This URL would look like this:

meridium://Registry/RecordManager

To configure the URL to open the record that is associated with the clicked datapoint, after the text **RecordManager**, you would need to type a question mark (?), followed by the EntityKey parameter. Then, click the **Special Token** button, and then click **Entity Key for Clicked Point**.

This step will automatically insert the text *[EntityKey]* into the URL.

The complete URL would look like this:

meridium://Registry/RecordManager?EntityKey=[EntityKey]

When you click a datapoint on the graph:

- The URL will be executed.
- The record that is associated with the datapoint that you clicked will appear in the Record Manager.

Example: Passing Datapoint Values to an Internal URL

The following example explains how to create a hyperlink such that when you click a datapoint on the graph, a new record will be opened in the Record Manager and the Asset Manufacturer field will be populated automatically with the name in the Asset Manufacturer in the record whose datapoint you clicked on the graph.

> Œ Hint: You can also pass a datapoint value into a URL that accesses an external website.

This example assumes that the graph is based on a query that returns the results shown in the following image.

Asset ID	Operating Temperature	Asset Manufacturer
HE-2931	400	METALFORMS
HE-2448	450	EXELL, INC
HE-1666	400	ALCO
HE-2919	400	SLAGLE MFG
HE-2554	450	OHMSTEDE,
HE-2445	450	OHMSTEDE,

Note that the Asset ID, Operating Temperature, and Asset Manufacturer fields are included in the query.

After the graph has been built from the query, it would look like the following image, assuming that you plot the Asset ID along the x-axis and the Operating Temperature along the y-axis.

Suppose all of the records in this graph belong to the Shell and Tube Heat Exchanger family, and you order all Shell and Tube Heat Exchangers from the same manufacturer. You might want to create a new Shell and Tube Heat Exchanger record from this graph and populate the Asset Manufacturer field automatically with the manufacturer that supplies your Shell and Tube Heat Exchangers.

To do so, you can create a hyperlink that allows you to click any datapoint in the graph to create a new Shell and Tube Heat Exchanger record where the Asset Manufacturer field is populated with the value in the Asset Manufacturer field in the record that is associated with the clicked datapoint.

To create this hyperlink, on the **Graph Setup** window, for the field that you are displaying as the series, you would need to add a URL that will open a new record in the Record Manager. This URL would look like this:

meridium://Registry/RecordManager?EntityKey=0&FamilyKey=[FamilyKey]

...where:

- The **EntityKey** parameter value is **0**, which creates a new record.

  -and-

- The **FamilyKey** parameter value is **[FamilyKey]**, which will create a record in the family of the datapoint you click on the graph.

> Œ**Hint:** You can let Meridium APM automatically insert the value *[FamilyKey]* into your URL by selecting **Family Key for Clicked Point** on the **Special Token** submenu.

So far, the URL will allow you to click a datapoint on the graph and create a new record in the same family as that of the record the clicked datapoint. If you want to populate a field in the new record automatically with the value in the record that is associated with the datapoint that you click, however, you will need to add another parameter to the URL: [FieldID].

To be able to pass the value from a clicked datapoint to the [FieldID] parameter, that field must be included in the graph's underlying query or dataset. In this example, we want to populate the Asset Manufacturer field automatically, and that field is included in the query.

To add the [FieldID] parameter to the URL, after the last parameter value, which in this case is [FamilyKey], you would need to type the field ID of the field that you want to populate, followed by the equal sign (=). Then, click the **Special Token** button, click **Available Column Values for Clicked Point**, and click **Asset Manufacturer** on the submenu.

This step will automatically insert the text *[Asset Manufacturer]* into the URL. So, the complete URL would look like this:

meridium://Registry/RecordManager?EntityKey=0&FamilyKey=[FamilyKey]&ASSET_ MANUF_CHR=[Asset Manufacturer]

When you click a datapoint on the graph:

- The URL will be executed.
- A new Shell and Tube Heat Exchanger record will appear in the Record Manager.
- The Asset Manufacturer field will be populated automatically with the value in the Asset Manufacturer field in the record whose datapoint you clicked on the graph.

Example: Passing Datapoint Values to an Internal URL

	Value(s)
Asset ID	
Asset Description	
Asset Type	
Asset Manufacturer	ALCO
Asset Model Number	
Asset Serial Number	
Asset Installation Date	
Asset Status	
Asset Service	
Critical Asset ?	☐
Spared ?	☐
Asset P&ID Drawing Number	
Shutdown Req. for Maintenance?	☐
Asset Additional Information	⋯

Example: Passing Datapoint Values to an External URL

The following example explains how to create an external URL to the Google™ search engine and automatically refine the search so that it returns results related to the manufacturer associated with the datapoint that you clicked on the graph.

This example assumes that the graph is based on a query that returns the results shown in the following image.

Asset ID	Operating Temperature	Asset Manufacturer
HE-2931	400	METALFORMS
HE-2448	450	EXELL, INC
HE-1666	400	ALCO
HE-2919	400	SLAGLE MFG
HE-2554	450	OHMSTEDE,
HE-2445	450	OHMSTEDE,

Note that the Asset ID, Operating Temperature, and Asset Manufacturer fields are included in the query.

After the graph has been built from the query, it would look like the following image, assuming that you plot the Asset ID along the x-axis and the Operating Temperature along the y-axis.

Suppose the pieces of equipment associated with each datapoint are manufactured by different companies. You could add a hyperlink to the graph so that when you click any of the datapoints, a Google™ search would appear, and the search would automatically return results related to the manufacturer associated with the clicked datapoint.

For example, if HEX1 is manufactured by Atlantic, and HEX4 is manufactured by Pacific, you could add a hyperlink so that when you click the datapoint for HEX1, the search results would be related to Atlantic. Likewise, when you click the datapoint for HEX4, the search results would be related to Pacific.

To add this hyperlink, for the field that you are displaying as the series, you would need to add a URL that will open the Google™ search engine. This URL would look like this:

http://www.google.com/search?

So far, the URL will allow you to click a datapoint on the graph and launch a new Google™ search. If you want to refine the search to return results associated with a specific manufacturer, however, you will need to add another parameter to the URL and define its value as the value in the Asset Manufacturer field in the record that is associated with the clicked datapoint.

To be able to pass the value from a clicked datapoint to the URL, that field must be included in the graph's underlying query or dataset. In this example, we want to search for a specific manufacturer, and the Asset Manufacturer field is included in the query.

In order to pass values into an external URL, you must already know the parameters that the URL accepts. In this example, since we are using a Google™ URL, we know that it accepts the parameter q, which lets you define the search criteria.

This means that before we can pass values into the URL, we need to add the q parameter to the URL. So, the URL we need to configure should now look like this:

http://www.google.com/search?q=

To provide an Asset Manufacturer value from the graph to the q parameter, on the **Graph Setup** window, after the last parameter value, which in this case is q, you would need to click the **Special Token** button, point to **Available Column Values for Clicked Point**, and then click **Asset Manufacturer**.

This step will automatically insert the text **<Asset Manufacturer>** into the URL. So, the complete URL would look like this:

http://www.google.com/search?q=<Asset Manufacturer>

When you click a datapoint on the graph:

- The URL will be executed.
- The Google™ website will appear.
- The search will return only results that are associated with the value in the Asset Manufacturer field in the record whose datapoint you clicked on the graph.

Note that you can extend this example to meet your company's needs. As long as you know the parameters that the external URL accepts, you can pass values from the graph to those parameters as desired.

About Specifying the Value That will be Displayed for a Datapoint

Graphs can be configured to show the datapoint values. In these cases, you will need to specify *which* datapoint values should be displayed on the graph. You can do so using the **Choose which value will be displayed as a point label** list on the **Graph Setup** window. The following options are available:

- **Y Axis value:** Displays the y-axis value.
- **Value and Category:** Displays the y-axis value and the category name, formatted as *y-axis value (Category Name)*. This option is available only for Category charts (pie, pyramid, doughnut, or radar).
- **Percent Value:** Displays the y-axis value as a converted percentage. This option is available only for Category charts (pie, pyramid, doughnut, or radar).
- **<Field Value>:** Displays the value that exists in the underlying dataset column or query field that is represented by the <Field Value>.

An Example of Showing the Y-Axis Value

Suppose that a graph is based on a query whose results look like those shown in the following image.

Work History Count	Taxonomy Category	Category Description
9311	FIX	Fixed Asset
6793	ROT	Rotating Asset
2979	INS	Instrument Asset
1271	ELE	Electrical Asset
172	SUB	Subcomponent
75	MAT	Material Handling Asset
67	FAC	Facility Asset
34	MIS	Miscellaneous Asset
6	OTH	Other Asset

These query results indicate the number of Work History records that are linked to Equipment records with each Taxonomy Category and Category Description values. For example, you can see that *9311* Work History records are linked to an Equipment record with the value *FIX* in the Taxonomy Category field and *Fixed Asset* in the Category Description field.

If you were to create a graph based on this query and show the *y-axis values* for the data-point labels, the graph would look like the one shown in the following image.

An Example of Showing the Y-Axis Value

Bad Actors by Count - Taxonomy Category Graph
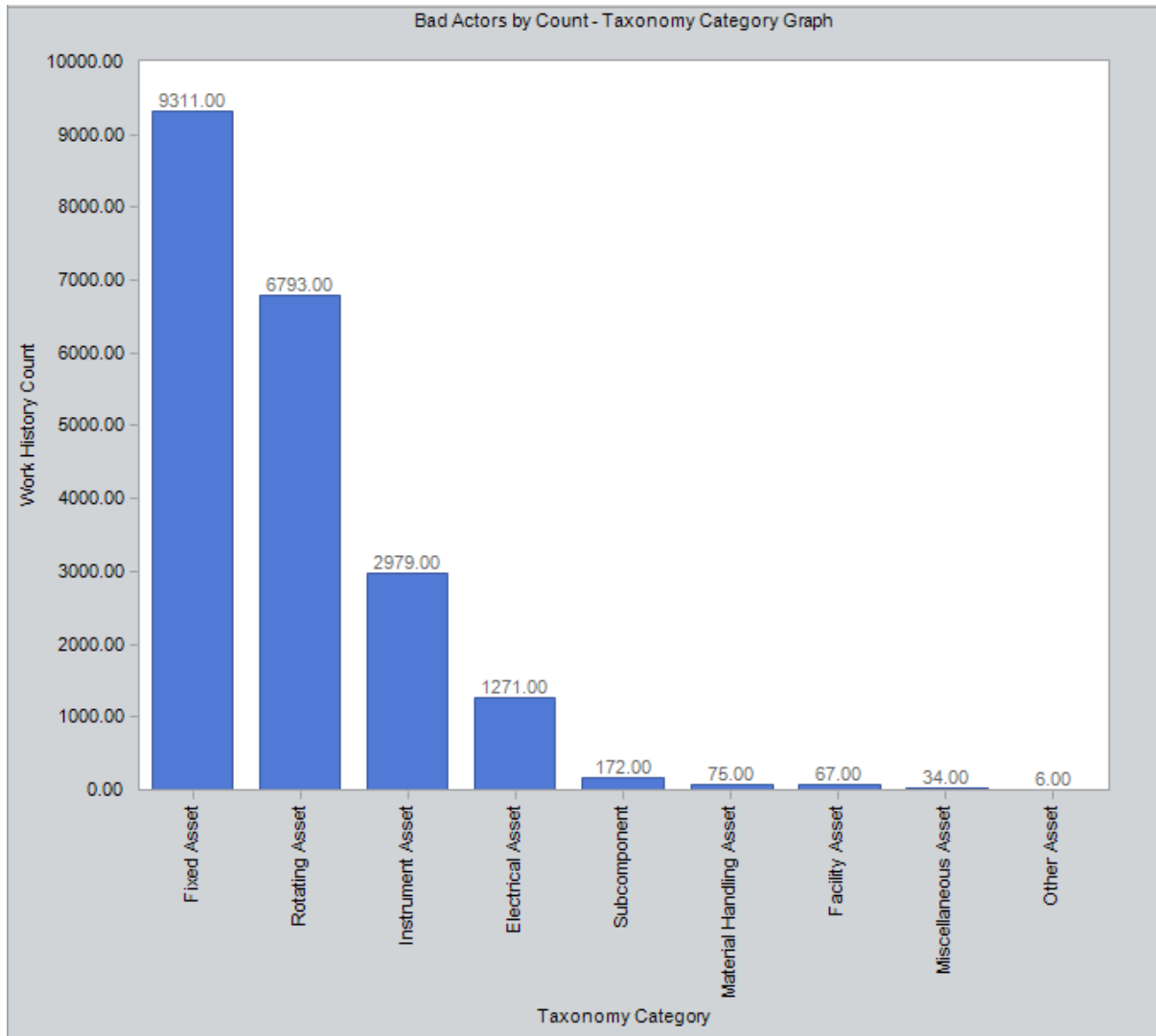
An Example of Showing the Value and Category

Suppose that a graph is based on a query whose results look like those shown in the following image.

Work History Count	Taxonomy Category	Category Description
9311	FIX	Fixed Asset
6793	ROT	Rotating Asset
2979	INS	Instrument Asset
1271	ELE	Electrical Asset
172	SUB	Subcomponent
75	MAT	Material Handling Asset
67	FAC	Facility Asset
34	MIS	Miscellaneous Asset
6	OTH	Other Asset

These query results indicate the number of Work History records that are linked to Equipment records with each Taxonomy Category and Category Description value. For example, you can see that *9311* Work History records are linked to an Equipment record with the value *FIX* in the Taxonomy Category field and *Fixed Asset* in the Category Description field.

If you were to create a graph based on this query and show the *Value and Category* for the datapoint labels, the graph would look like the one shown in the following image.

In this example, you can see that for each series, the Work History Count value is displayed along with the corresponding Category Description value. For example, for the largest series (colored peach), you can see the value *9311.00 (Fixed Asset)*. You can determine, therefore, that *9311* Work History records are linked to an Equipment record with the Category Description *Fixed Asset*.

If you were to view the graph configuration, you would see that the *Category Description* field is selected as the *X Axis/Category*, as shown in the following image.

Choose columns to display in graph :

☐ Include Hyperlink

Special Token >

☐ Display **Taxonomy Category**

as
- ○ X Axis / Category
- ○ Series
- ○ Constant (X Axis)
- ○ Constant (Y Axis)

☑ Display **Category Description**

as
- ⊙ X Axis / Category
- ○ Series
- ○ Constant (X Axis)
- ○ Constant (Y Axis)

An Example of Showing the Percent Value

Suppose that a graph is based on a query whose results look like those shown in the following image.

Work History Count	Taxonomy Category	Category Description
9311	FIX	Fixed Asset
6793	ROT	Rotating Asset
2979	INS	Instrument Asset
1271	ELE	Electrical Asset
172	SUB	Subcomponent
75	MAT	Material Handling Asset
67	FAC	Facility Asset
34	MIS	Miscellaneous Asset
6	OTH	Other Asset

These query results indicate the number of Work History records that are linked to Equipment records with each Taxonomy Category and Category Description value. For example, you can see that *9311* Work History records are linked to an Equipment record with the value *FIX* in the Taxonomy Category field and *Fixed Asset* in the Category Description field.

If you were to create a graph based on this query and show the legend and *Percent Value* for the datapoint labels, the graph would look like the one shown in the following image.

In this example, you can see that for each series, the percentage of Work History records is displayed. For example, for the largest series (colored peach), you can see the value *44.96%*. You can determine, therefore, that *44.96%* of existing Work History records are linked to an Equipment record with the Category *Fixed Asset*.

An Example of Showing the Field Value

Suppose that a graph is based on a query whose results look like those shown in the following image.

Work History Count	Taxonomy Category	Category Description
9311	FIX	Fixed Asset
6793	ROT	Rotating Asset
2979	INS	Instrument Asset
1271	ELE	Electrical Asset
172	SUB	Subcomponent
75	MAT	Material Handling Asset
67	FAC	Facility Asset
34	MIS	Miscellaneous Asset
6	OTH	Other Asset

These query results indicate the number of Work History records that are linked to Equipment records with each Taxonomy Category and Category Description value. For example, you can see that *9311* Work History records are linked to an Equipment record with the value *FIX* in the Taxonomy Category field and *Fixed Asset* in the Category Description field.

If you were to create a pyramid graph based on this query and show the value in the *Taxonomy Category* field for the datapoint labels, the graph would look like the following image.

In this example, you can see that for each series, the Taxonomy Category value is displayed. For example, for the largest series (colored blue), you can see the value *FIX*. You can determine, therefore, that the majority of Work History records are linked to an Equipment record with the Taxonomy Category *FIX*.

If you were to view the graph configuration, you would see the value *TaxonomyCategory* in the **Choose which value will be displayed as a point label** list, as shown in the following image.

Overview of the Graph Viewer Page

After you create a new graph or open an existing graph from the Catalog, it appears on the **Graph Viewer** page. The **Graph Viewer** page contains a workspace, which is the main display area to the right of the task menus.
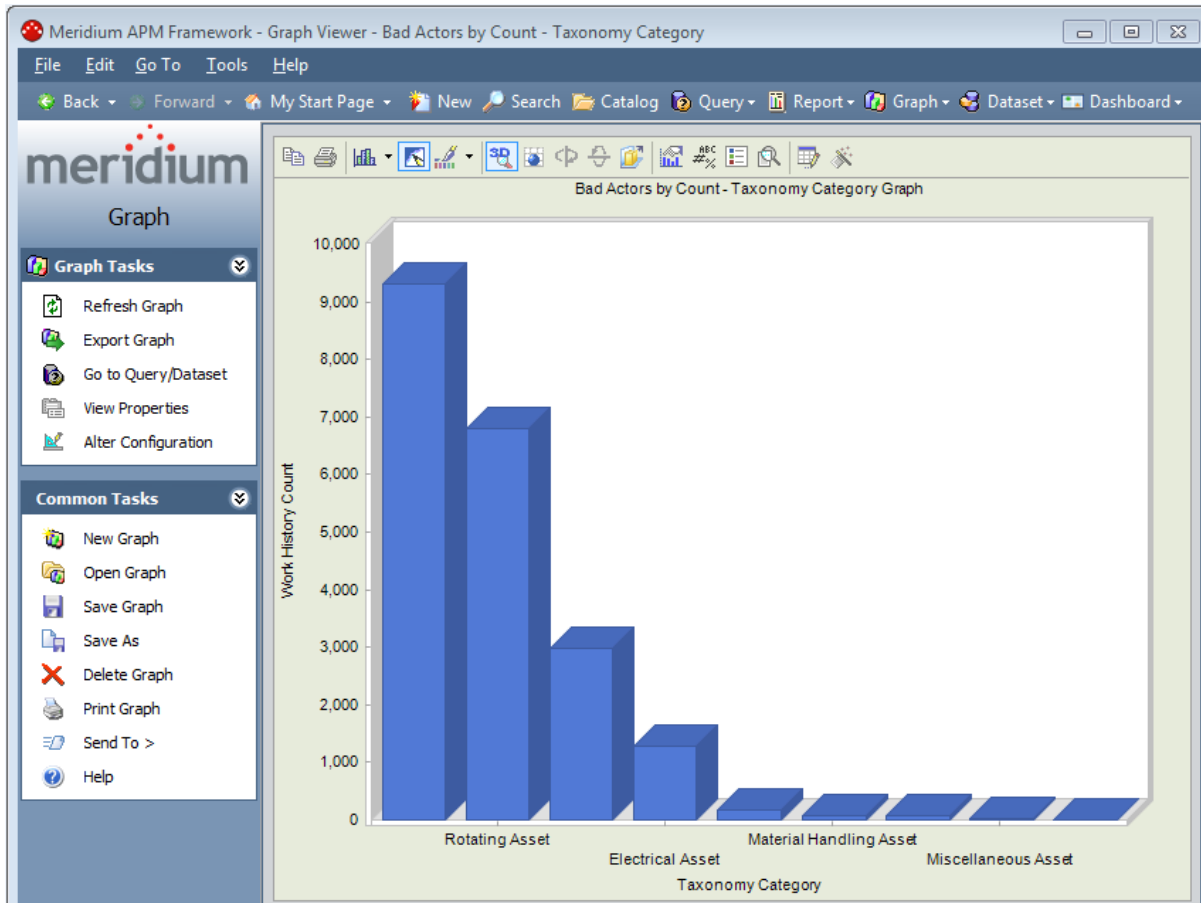
Aspects of the Graph Workspace

When you open a graph, the graph itself is displayed in the graph workspace, which consists of the following areas:

- **Toolbar:** The area that appears at the top of the graph, displaying options that let you modify the appearance of the graph. Note that the standard toolbar appears by default.
- **Background:** The area surrounding the graph, where the border and axis labels appear.
- **Chart Box:** The area in the graph where the gridlines and datapoints appear.
- **Series:** The area on the graph where the values from the underlying query or dataset are represented by datapoint markers (e.g., bars, lines, bubbles).

The following image shows a graph as it appears on the **Graph Viewer** page. Each main area of the graph workspace is labeled accordingly.
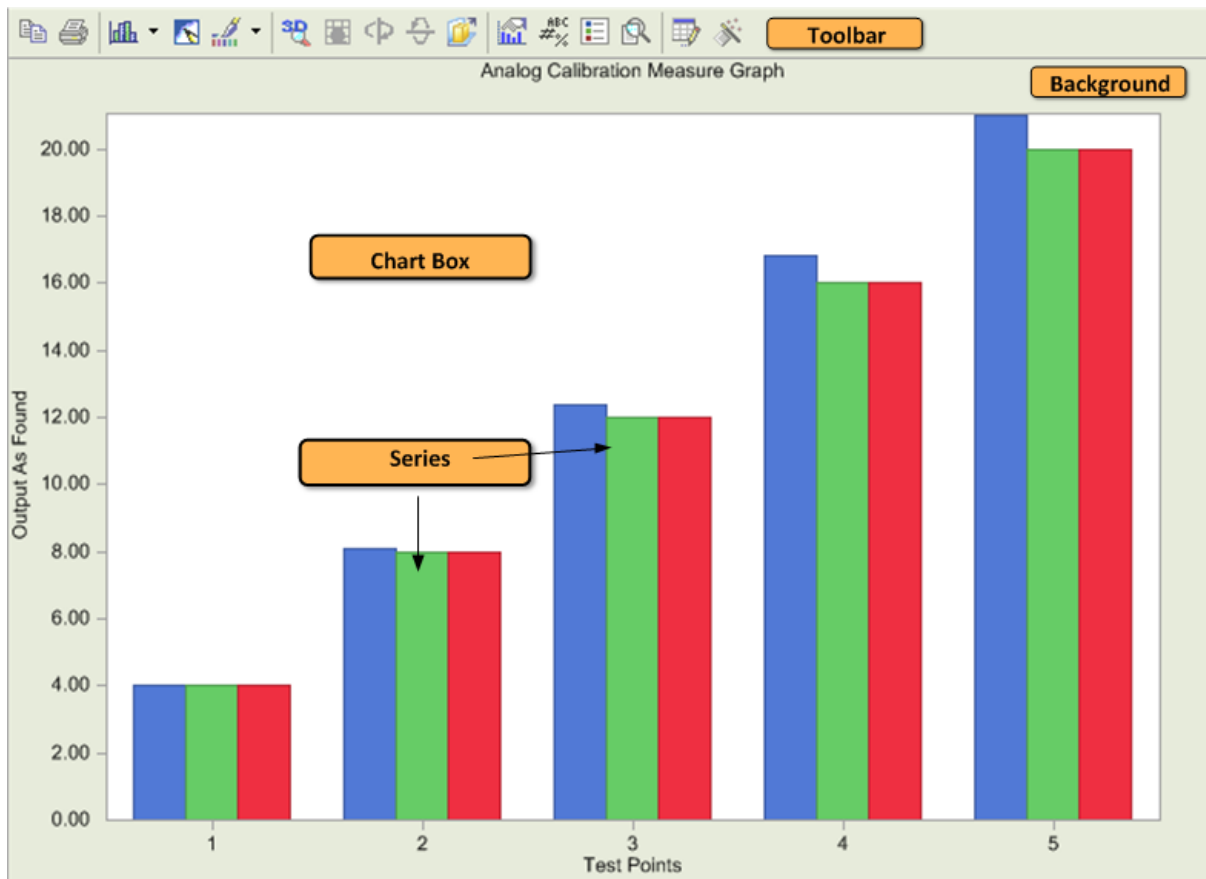
Standard Toolbar

A standard toolbar appears at the top of a graph and displays options that let you change the appearance of the graph.

The following table provides a description of each button that appears on the standard toolbar.

Button	Name	Description
	Copy	Displays a submenu from which you select the format in which you want to copy the graph: **As a Bitmap**, **As a Meta-file**, or **As Text (data only)**.
	Print	Displays the **Print** dialog box, from which you can print the graph.
	Gallery	Displays a drop-down menu of graph types. You can click any option to change the graph type.
	Anti-Ali-asing	Applies or removes anti-aliasing, whichever is the opposite of the current setting. When anti-aliasing is enabled, the lines on the graph and text appear smoother.
	Palette Selector	Displays a drop-down list of color schemes. You can click a color scheme to modify the colors of the graph.
	3D/2D	If the graph that you are currently viewing is displayed as a two-dimensional figure, clicking the **3D/2D** button will display the graph as a three-dimensional figure, and vice versa.
	Rotated View	If you are viewing a 3D figure, clicking this option converts the graph to *Rotated View*. After switching to *rotated view*, you can click the ⊕ or ⊕ buttons to rotate the graph to the desired angle.
	Rotate Around Y Axis	In the rotated view, rotates the graph on the y-axis.
	Rotate Around X Axis	In the rotated view, rotates the graph on the x-axis.

Button	Name	Description
	Clustered (Z Axis)	In graphs that contain two or more series, stacks one series in front of the other(s) so that you can compare the results more easily.
	Axes Settings	Displays a drop-down menu of options that let you modify the appearance of the y-axis and the x-axis.
	Point Labels	Inserts labels on the graph so that you can see the underlying value for each point as it appears in the underlying query or dataset.
	Legend Box	Displays the legend that indicates the meaning of each color in the graph.
	Zoom	Lets you zoom in on a selected area of the graph. To zoom in, click the Zoom button, and then click the area that you want to enlarge. To re-display the graph in its original scale, click the Zoom button again.
	Data Editor	Displays the Data Editor, which shows the underlying values associated with the graph you are currently viewing.
	Properties	Displays the Properties dialog box, where you can modify the appearance of the graph.

Annotation Toolbar

The annotation toolbar does not appear on the **Graph Viewer** page by default. To display the annotation toolbar, right-click anywhere on the graph, and on the shortcut menu, point to **Toolbars**, and then click **Annotation Toolbar**. The annotation toolbar will appear below the standard toolbar.

The annotation toolbar offers options that let you customize the appearance of the graph by adding annotation objects to the graph. The following table provides a description of each button that is available on the annotation toolbar.

Button	Name	Description
	Selector	Lets you select any shapes, text boxes, or images that you have inserted.
	Rectangle	Lets you draw a rectangle on the graph.
	Circle	Lets you draw a circle on the graph.
	Arrow	Lets you draw an arrow on the graph.
	Arc	Lets you draw an arc on the graph.
	Picture	Lets you choose an area to insert an image.
	Text	Lets you create a text box, where you can type new or modify existing text.
	Balloon	Lets you create a balloon-shaped text box, where you can type new or modify existing text.
	Polygon	Lets you draw an asymmetric shape.
	Fill Color	Displays a drop-down list of color schemes. You can click a color scheme to modify the fill color for any rectangle or circle.
	Line Color	Displays a drop-down list of color schemes. You can click a color scheme to modify the colors of the lines around any shape.
	Copy	Lets you copy an object on the graph.

Button	Name	Description
	Paste	Allows you to paste into the graph any object that has been copied.
	Bring to Front	Lets you arrange any shapes, text boxes, or images that are staggered by bringing the selected item to the front.
	Send to Back	Lets you arrange any shapes, text boxes, or images that are staggered by sending the selected item to the back.
	Group	Allows you to group selected objects together.
	Ungroup	Allows you to ungroup objects that are grouped together.
	Flip Vertical	Flips the selected item vertically.
	Flip Horizontal	Flips the selected item horizontally.
	Rotate Right	Rotates the selected item to the right.
	Rotate Left	Rotates the selected item to the left.

Graph Tasks Menu
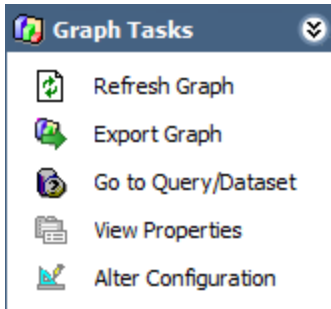
The **Graph Tasks** menu on the **Graph Viewer** page contains the following links:

- **Refresh Graph:** Reloads the graph and shows the most current information, including recent changes to the configuration options or changes to the underlying query or dataset.

- **Export Graph:** Displays the **Export Graph to File** dialog box, which lets you export the graph to an external location.

- **Go to Query/Dataset:** Displays the query or dataset on which the graph is based.

- **View Properties:** Displays the **Graph Properties** dialog box, where you can view the information associated with the graph, including the file name, location, and last modified date.

- **Alter Configuration:** Displays the **Graph Setup** window, where you can change the configuration options associated with the graph.

Common Tasks Menu
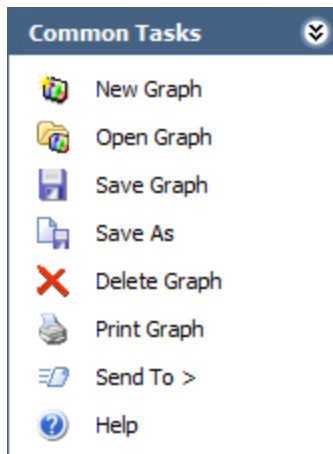
The **Common Tasks** menu on the **Graph Viewer** page contains the following links:

- **New Graph:** Displays the **Choose a Query or Dataset for Graph** dialog box, where you can create a new graph.
- **Open Graph:** Displays the **Open existing Graph** dialog box, where you can select an existing graph and then open it.
- **Save Graph:** Saves the graph that you are currently viewing.
- **Save As:** Displays the **Save Graph As** dialog box, where you can save the current graph with a different name or to a different location in the Catalog.
- **Delete Graph:** After asking for confirmation, deletes the graph that you are currently viewing.
- **Print Graph:** Displays the **Print** dialog box, from which you can print the graph that you are currently viewing.
- **Send To:** Displays a submenu with options that let you provide a link to the current page on your desktop (create shortcut), in an email message, or on a Home Page.
- **Help:** Displays the context-sensitive Help topic for the **Graph Viewer** page.

Accessing the Properties Dialog Box

The **Properties** dialog box is accessible from the **Graph Viewer** page and provides access to options that allow you to modify the appearance of the graph.

To access the Properties dialog box:

- In the Meridium APM Framework, on the **Graph Viewer** page, on the standard tool-bar, click the **Properties** button.

  -or-

  Right-click anywhere on the graph, and then on the shortcut menu, click **Prop-erties**.

  The **Properties** dialog box appears, displaying the current properties that are applied to the graph.
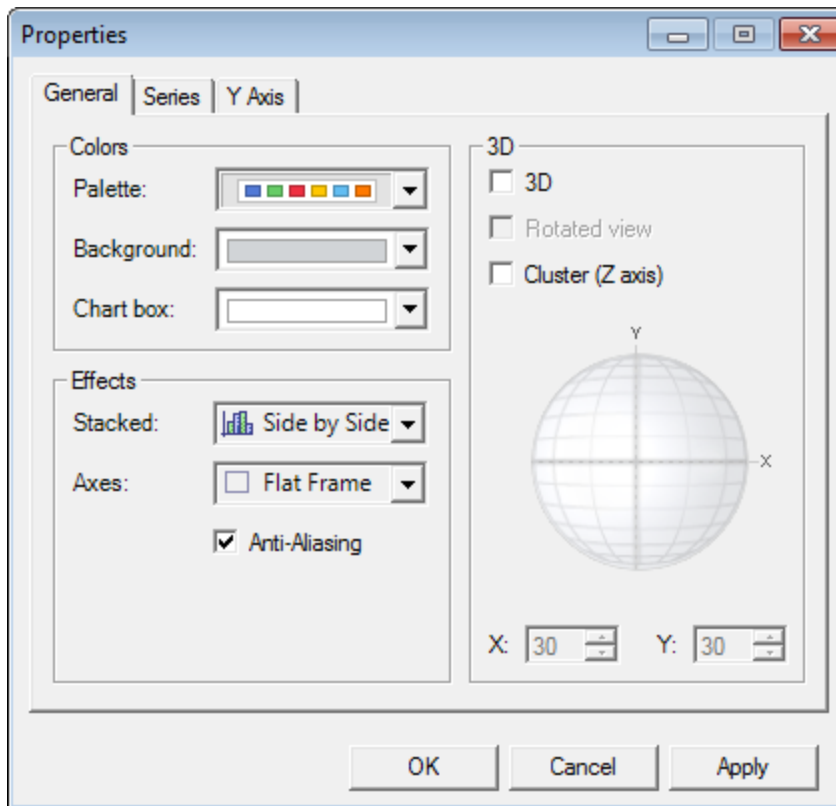
> Note: When you access the **Properties** dialog box in this way, the **Y Axis** tab applies to the *left* y-axis. If your graph contains a *right* y-axis, you can modify its properties by right-clicking the *right* y-axis, and choosing **Properties**. The **Properties** dialog box shown above will appear, but the **Y Axis** tab will then apply to the *right* y-axis. Similarly, you can access the properties of the x-axis by right-clicking the x-axis and then choosing **Properties**.

About the Properties Dialog Box

The **Properties** dialog box contains three tabs:

- **General**
- **Series**
- **<Axis Type> Axis**, where **<Axis Type>** is the value *X, Y,* or *Z* depending upon the type of graph that you are viewing and how you accessed the **Properties** dialog box. Throughout this documentation, we refer to this tab simply as the *Axis tab*.
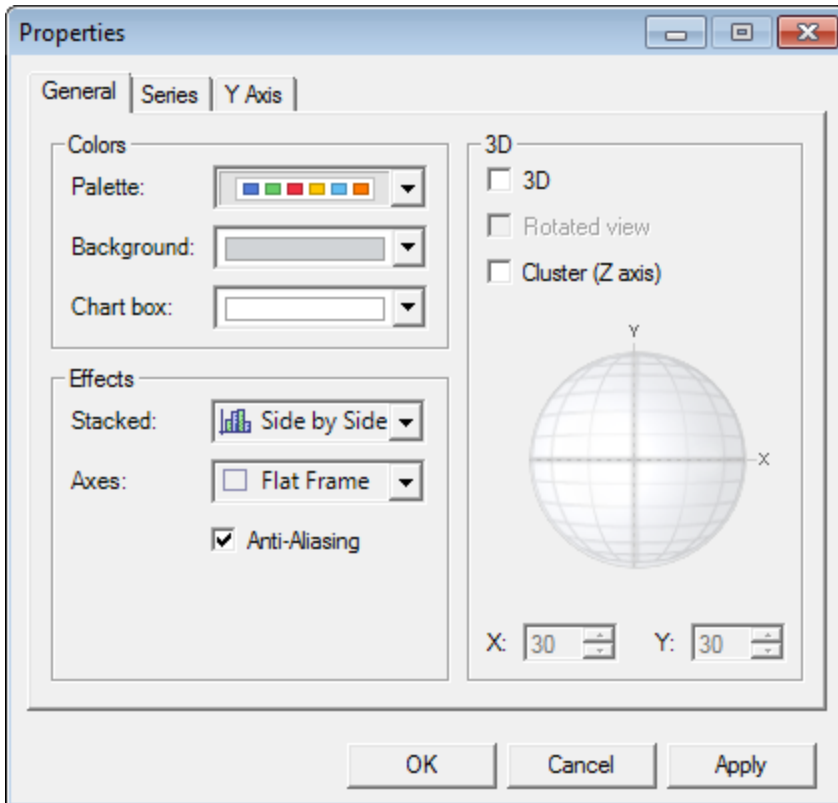
Each tab on the **Properties** dialog box contains sections that display the current property settings for the graph. The area of the graph affected by these settings is specified by the name on the tab. For example, the **General** tab contains settings for the overall appearance of the graph, including the color scheme.

Note that most of the settings that appear on the **Properties** dialog box can also be modified via the graph standard toolbar or the annotation toolbar.

The Colors Section

The **Colors** section on the **General** tab contains options that let you modify the [colors that are applied to the graph](). The following options are available:

- **Palette:** Displays color palettes that you can apply to all the [areas on the graph]() (i.e., background, chart box, and series). The color palettes consist of a set of six colors. For example, the following image shows the color palette list. Each row in the list is one color palette that you can select.

  After you select a color palette, the graph is displayed using the colors in the palette. For example, the following image shows a graph after a color palette has been applied to the graph.
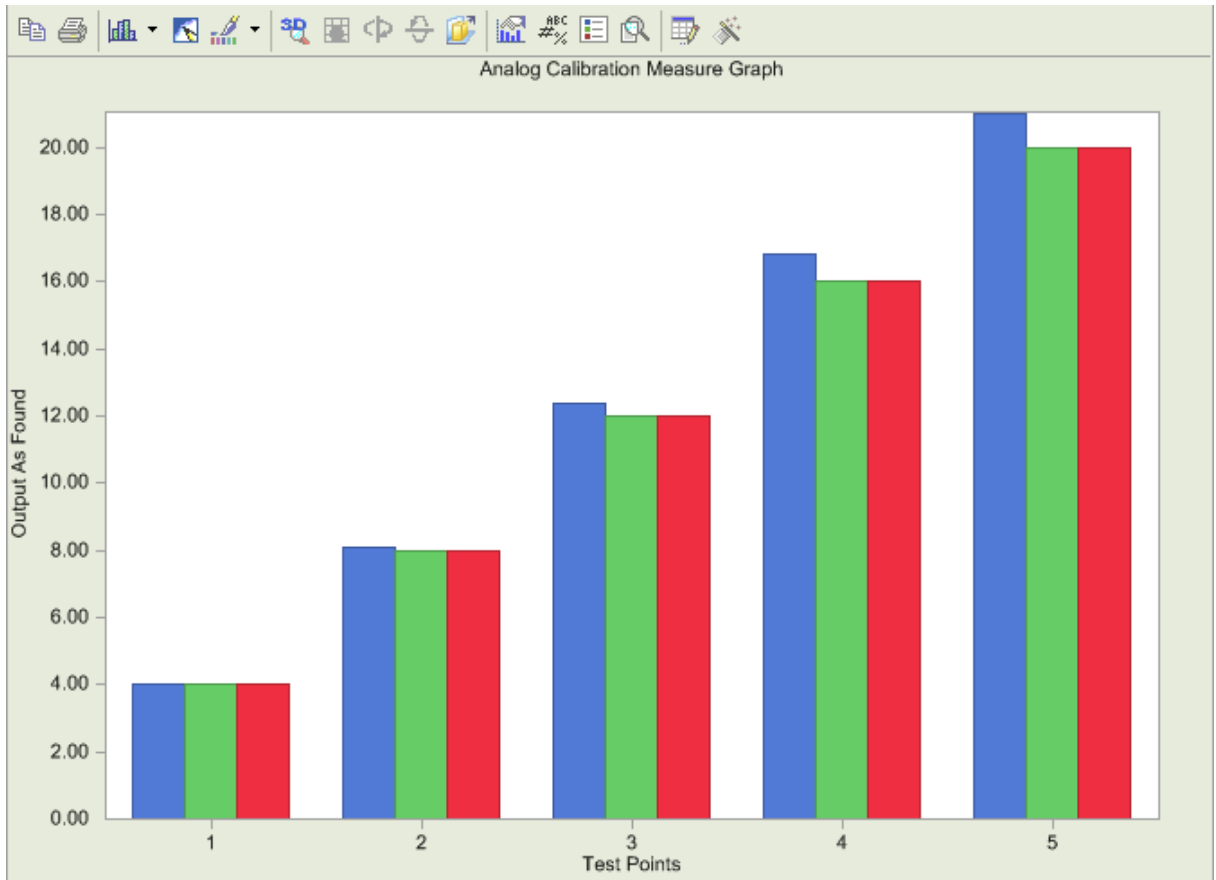
- **Background:** Displays colors that you can apply to the background area of the graph (including the toolbar area). For example, the following image shows a graph after a color has been applied to the background area.
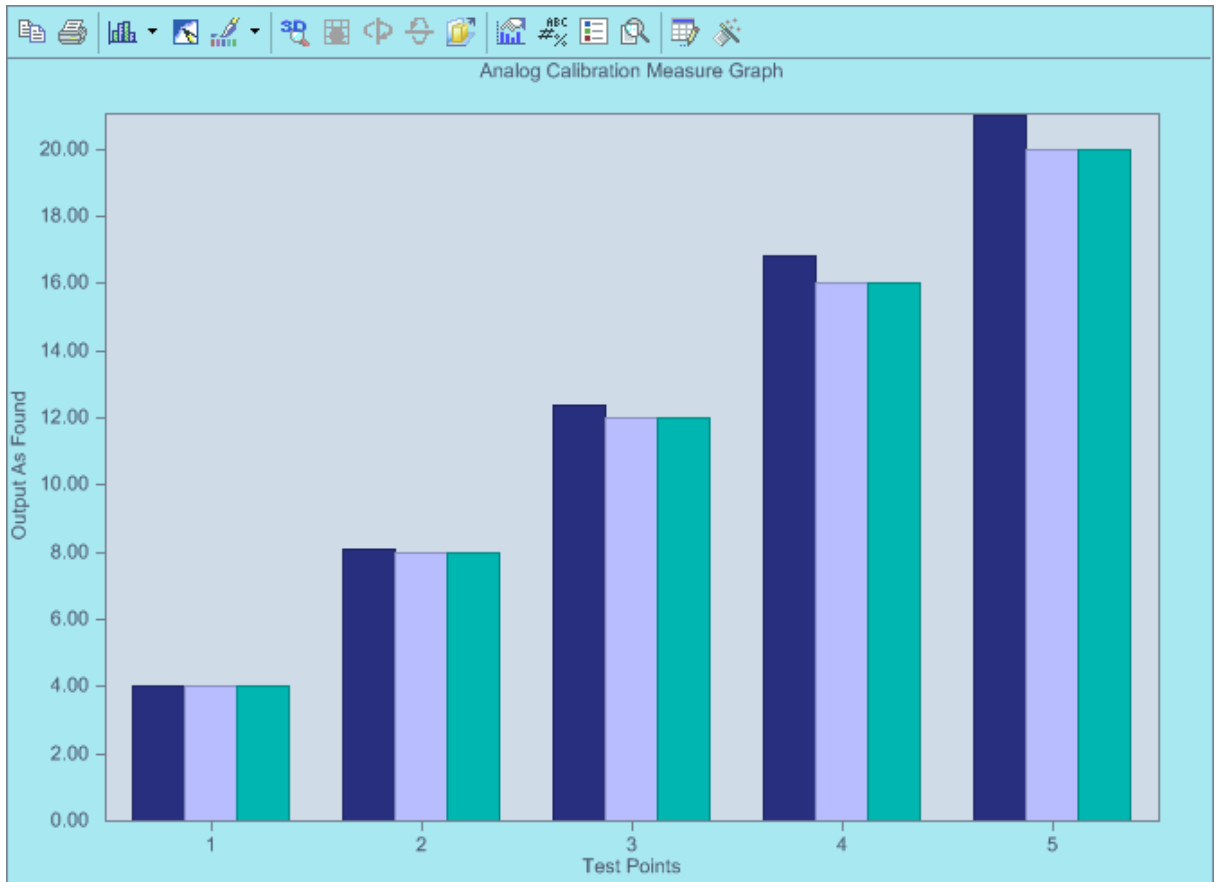
If a color palette is already applied to the graph, when you select a background color other than the one that is currently displayed, the chart box and series area colors will still be displayed using the selected color palette.

- **Chart Box:** Displays colors that you can apply to the chart box area of the graph. For example, the following image shows a graph after a color has been applied to the chart box area.
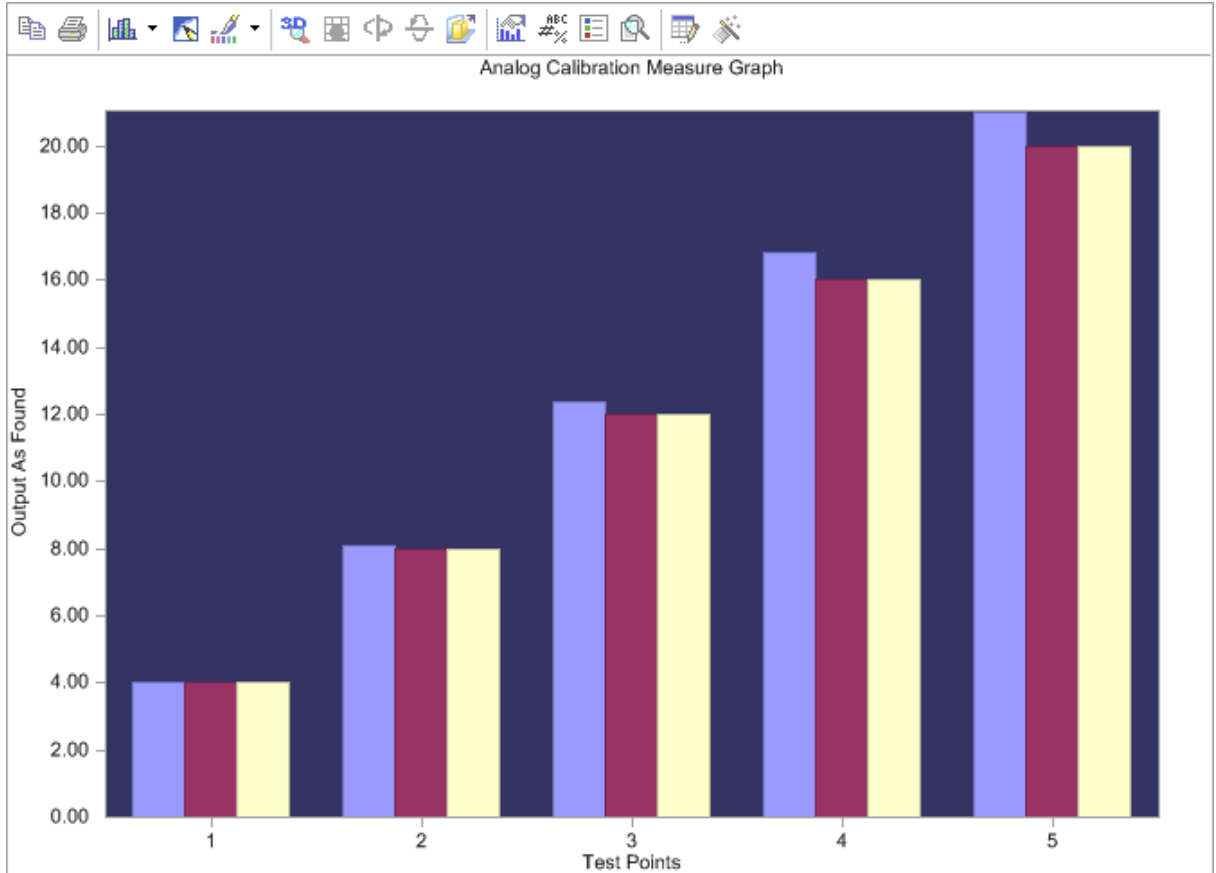
If a color palette is already applied to the graph, when you select a chart box color other than the one that is currently displayed, the background and series colors will still be displayed using the selected color palette.

The Effects Section

The **Effects** section on the **General** tab contains options that let you select the effects that are applied to the graph. The following options are available:

- **Stacked:** Lets you specify how you want the items in a series to be displayed. You can choose from the following options:
    - **Side by Side:** Displays the datapoint markers in the series area next to one another. The default setting for datapoint markers is **Side by Side**. For example, the following image shows a graph whose datapoint markers appear next to one another.

    - **Stacked:** In a graph that contains two or more columns or fields that are measured on the y-axis scale, combines datapoint markers whose values are the same and displays them on top of one another. For example, the following image shows a graph whose series are stacked. Note that this graph is also shown in a 3D format so that the colors are more distinguishable.

- **Stacked 100%:** In a graph that contains two or more columns or fields that are measured on the y-axis scale, groups the datapoint markers whose values are close together, providing a better view of those datapoint markers whose values are outliers. Note that the graph will adjust the scale to accommodate the data. For example, the following image shows the same graph in the previous example displayed using the **Stacked 100%** option.

- **Axes:** Lets you specify how the frame around the chart box area appears. You can choose from the following options:
    - **None:** Removes the frame from the graph (i.e., the outer edge of the chart box). For example, the following image shows a graph whose frame has been removed. Note that a line does not appear around the chart box area to distinguish the chart box area from the background area.

- **3D Frame:** Adds a three dimensional edge to the chart box area. For example, the following image shows a graph whose frame is displayed using the **3D Frame** option.

> **Note:** In the upper right corner of the frame, you can see the depth that is provided by the **3D Frame** option. Note also that the effect of this option is easier to distinguish when the graph is displayed using the 3D format.

- **Math:** Adds an edge along the x- and y-axis only.
- **Flat Frame:** Adds a line on all sides of the graph, framing the chart box area. Note that this option is the default option.

- **Anti-Aliasing:** A check box that you can select to make the lines on the graph appear smoother. For example, the following image shows a line graph where the **Anti-Aliasing** check box is cleared. Note that the lines appear jagged.

After you select the **Anti-Aliasing** check box on the same graph, the lines on the series appear smoother.

The **Effects** section on the **General** tab contains options that let you select the effects that are applied to the graph. The following options are available:

- **Stacked:** Lets you specify how you want the items in a series to be displayed. You can choose from the following options:
    - **Side by Side:** Displays the datapoint markers in the series area next to one another. The default setting for datapoint markers is Side by Side. For example, the following image shows a graph whose datapoint markers appear next to one another.

- **Stacked:** In a graph that contains two or more columns or fields that are measured on the y-axis scale, combines datapoint markers whose values are the same and displays them on top of one another. For example, the following image shows a graph whose series are stacked. Note that this graph is also shown in a 3D format so that the colors are more distinguishable.

- **Stacked 100%:** In a graph that contains two or more columns or fields that are measured on the y-axis scale, groups the datapoint markers whose values are close together, providing a better view of those datapoint markers whose values are outliers. Note that the graph will adjust the scale to accommodate the data. For example, the following image shows the same graph in the previous example displayed using the Stacked 100% option.

- **Axes:** Lets you specify how the frame around the chart box area appears. You can choose from the following options:
  - **None:** Removes the frame from the graph (i.e., the outer edge of the chart box). For example, the following image shows a graph whose frame has been removed. Note that a line does not appear around the chart box area to distinguish the chart box area from the background area.

- **3D Frame:** Adds a three dimensional edge to the chart box area. For example, the following image shows a graph whose frame is displayed using the **3D Frame** option.

> **Note:** In the upper right corner of the frame, you can see the depth that is provided by the 3D Frame option. Note also that the effect of this option is easier to distinguish when the graph is displayed using the 3D format.

- ▪ **Math:** Adds an edge along the x- and y-axis only.
- ▪ **Flat Frame:** Adds a line on all sides of the graph, framing the chart box area. Note that this option is the default option.

- **Anti-Aliasing:** A check box that you can select to make the lines on the graph appear smoother. For example, the following image shows a line graph where the **Anti-Aliasing** check box is cleared. Note that the lines appear jagged.

After you select the **Anti-Aliasing** check box on the same graph, the lines on the series appear smoother.

The 3D Section

The **3D** area on the **General** tab contains options that let you modify the way the graph appears. The following options are available:

- **3D:** A check box that you can select to display the graph in a three dimensional format. For example, the following image shows a graph that is displayed using the **3D** option.
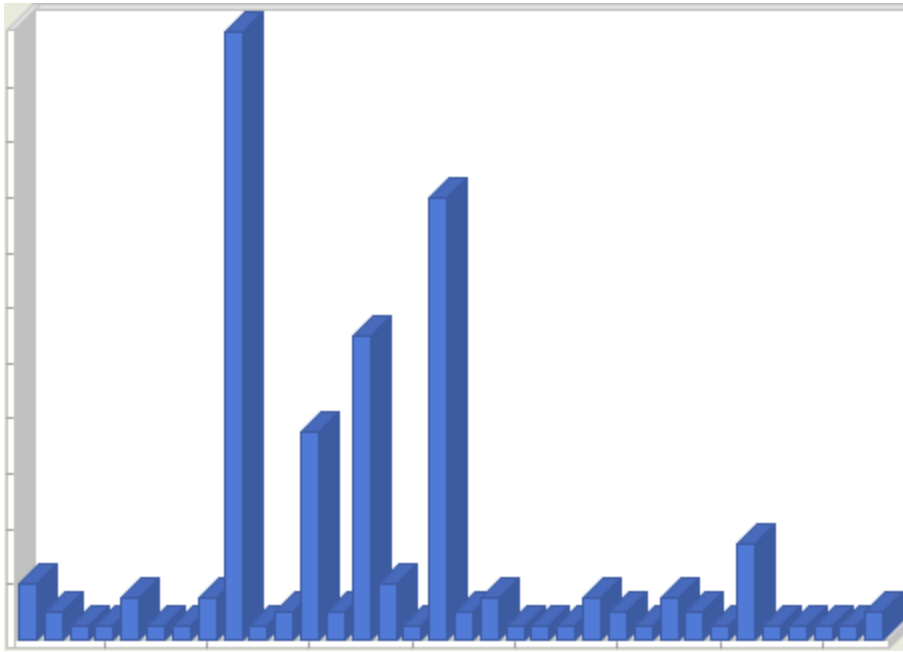
- **Rotated View:** A check box that you can select to display he graph in a Rotated View, meaning that you can display the graph at an *angle* that you specify. This check box is enabled only if the **3D** check box is selected.

  When the **Rotated view** check box is selected, the angle options are enabled, where you can move a pointer on the globe to the desired angle, or you can use the **X:** and **Y:** options to specify the desired angle. For example, the following image shows the **Properties** dialog box, displaying the angle options in the **3D** section. The angle options have been modified to display the graph at an angle of 50 degrees on the x-axis and y-axis.
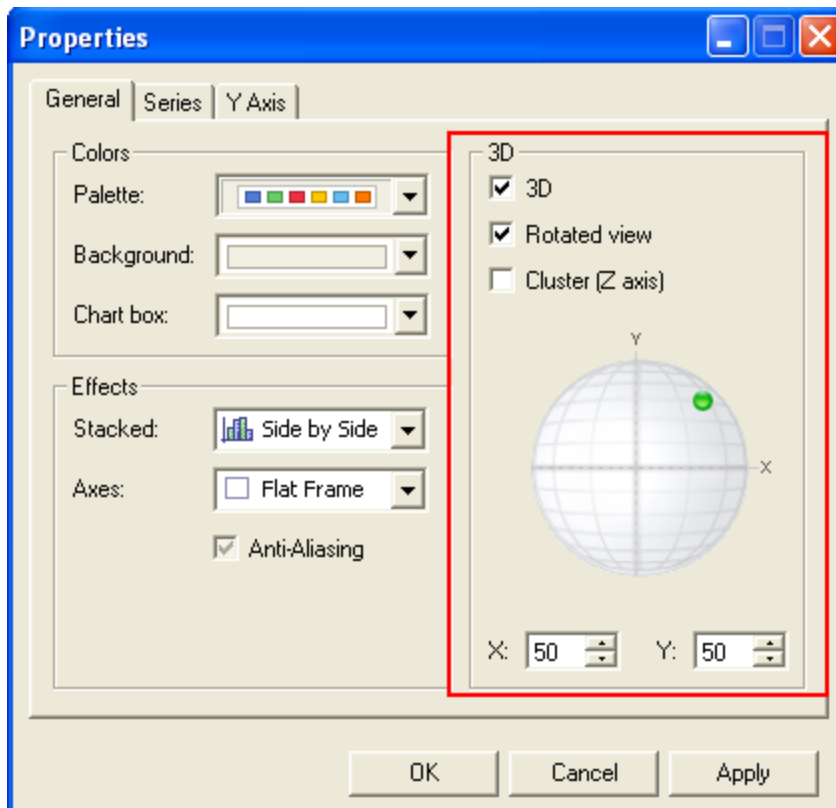
The following image shows a graph that is displayed using the 3D settings in the image shown above.
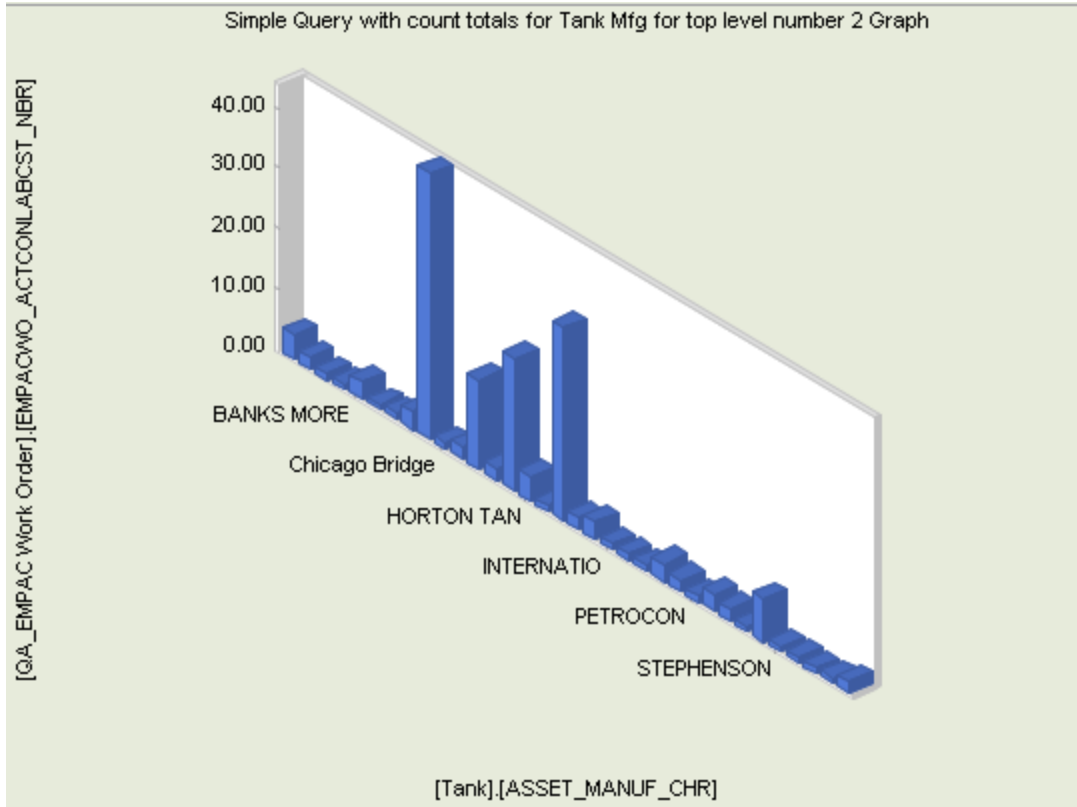
- **Cluster:** A check box that you can select to display multiple datapoint markers with the same values in the same location on the graph. The datapoint markers are displayed one behind the other. Note that while this option is enabled regardless of the status of the **3D** check box, if you want to see all the series in the graph when this option is selected, you should display the graph in 3D format. For example, a graph whose datapoint markers are displayed using the **Cluster** option *only* would look similar to this:
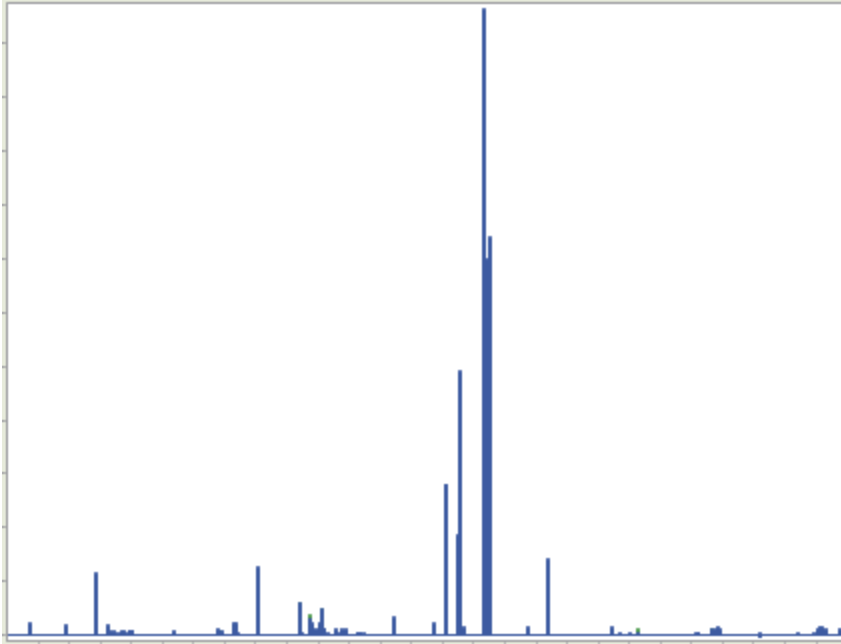
...and the same graph displayed using the **3D** option and the **Cluster** option would look similar to this:
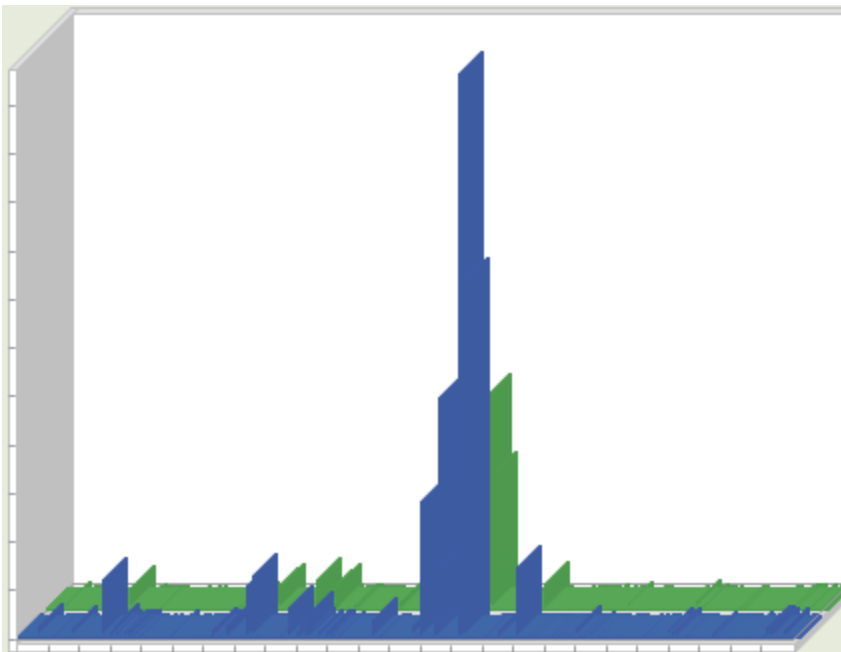
Note that, in the previous image, the green datapoint markers are now visible.

- **\X:** and **Y:** Let you specify the angle at which the graph will be displayed. You can also specify the angle by clicking and dragging the green pointer to the desired location on the globe. Note that when you move the pointer on the globe, the angle to which you adjust the pointer is displayed in the **X:** and **Y:** text boxes. These options are enabled only if the **Rotated view** check box is selected. For an

example of these options, see the description of the **Rotated view** option shown above.

The <All> Series Section

The **<All> Series** section on the **Series** tab contains the following options that let you choose the type of graph that is displayed and whether or not the underlying datapoint values are displayed. Note that *series* refers to a set of data that is measured on the y-axis and represented by datapoint markers that appear in the series area in a graph.

> **Note:** The **<All>** value identifies the series to which these settings will be applied. For example, if you right-click a series and access the **Properties** dialog box, the **Series** tab will display the name on the axis that is associated with the series that you clicked. Note also that accessing the **Properties** dialog box using the **Properties** button will always access the properties for ALL series. If you want to modify the properties of one, specific series, right-click the desired series, and click **Properties** on the shortcut menu.

The following options are available:

- **Gallery:** Lets you select the [type of graph](#) (e.g., bar and area graphs). For example, suppose you are viewing a bar graph and you want to view a line graph instead. You can select **Line** in the **Gallery** list, and then click the **Apply** button. The graph will refresh and display the data in a line graph. The following image shows the gallery list as it appears on the **Properties** dialog box.
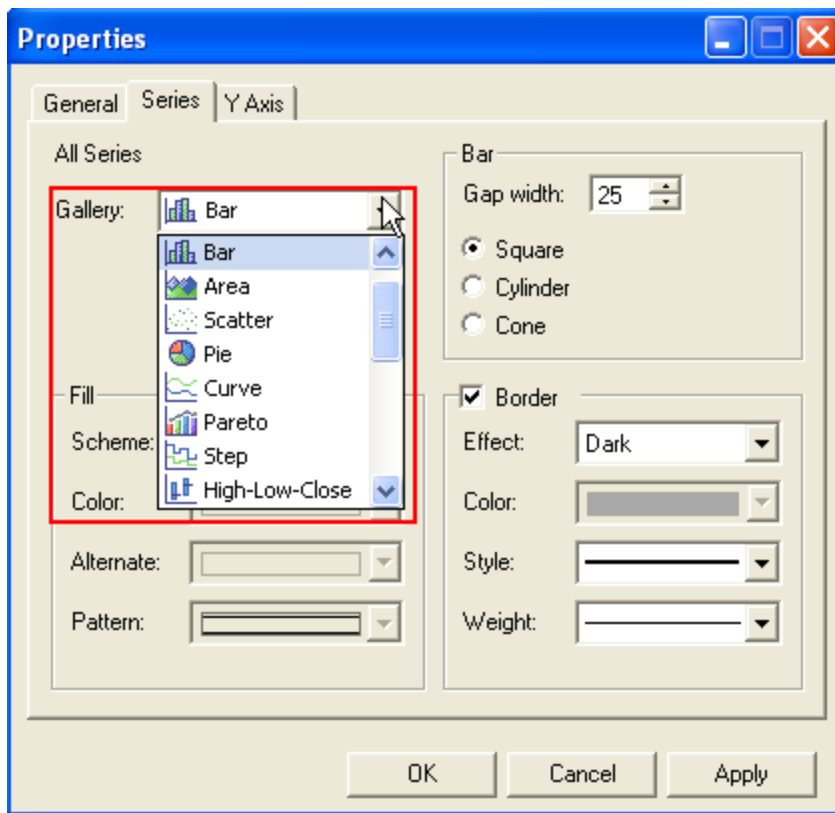
- **Point Labels:** A check box that you can select to display the values of the datapoint markers. For example, the following image shows a graph whose datapoint values are displayed.
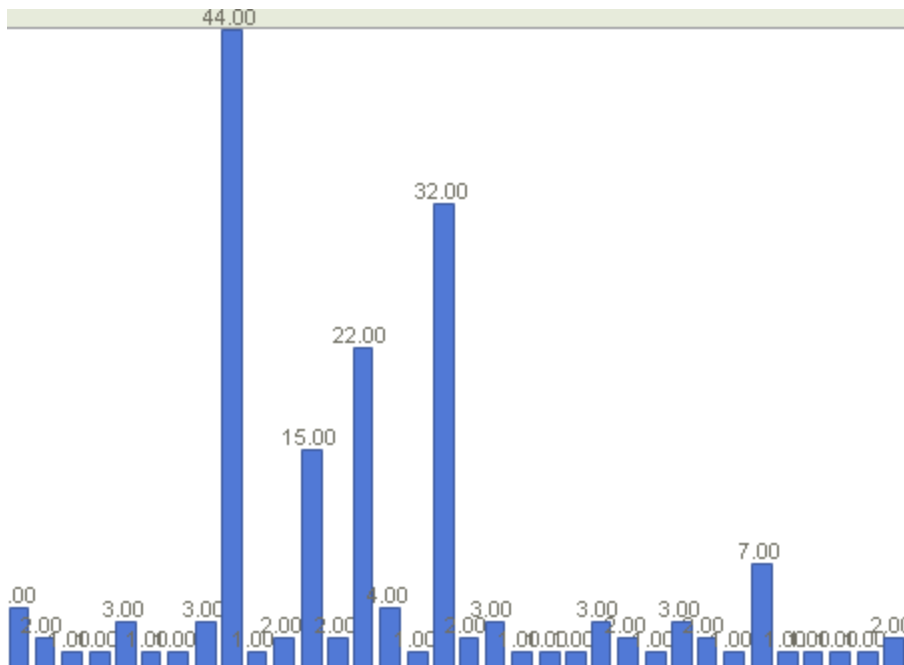
The Fill Section

The **Fill** section on the **Series** tab contains options that let you modify the appearance of the datapoint markers on the graph. Note that for these options to be enabled, you will need to right-click on a series marker and access the **Properties** dialog box. The following options are available:

- **Scheme:** Displays options that determine the appearance of the datapoint markers (e.g., bars, bubbles, and candlesticks). You can choose from the following:

  - **Solid:** Displays the datapoint markers using a solid color that you select from the **Color** list. For example, the following image shows a graph whose series markers are displayed in a solid color.
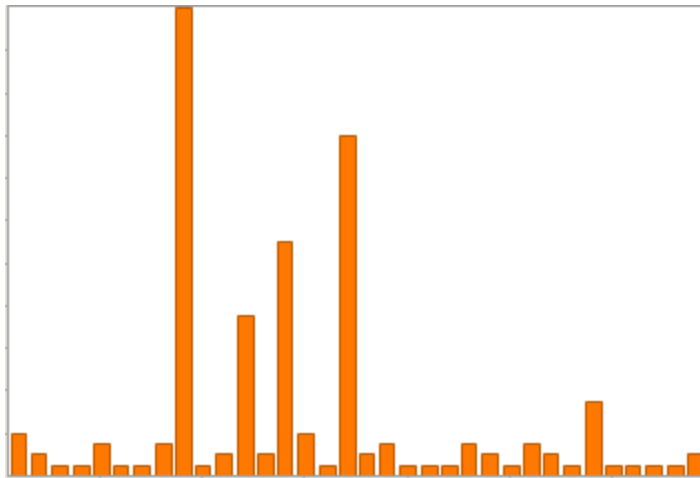
- **Monochrome:** Displays the datapoint markers in a black and white pattern that you select from the **Pattern** list. For example, the following image shows a graph whose datapoint markers have been modified using the **Monochrome** option and the **Pattern** list.
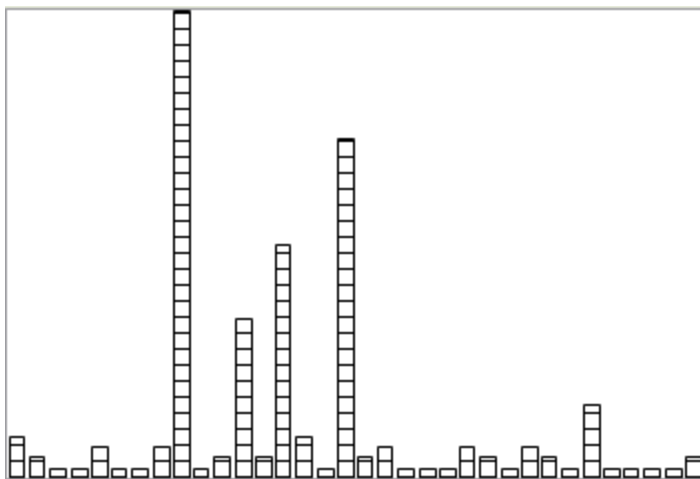
- **Patterns:** Displays the datapoint markers using the pattern and colors that you select in the **Pattern**, **Color**, and **Alternate** lists. For example, the following image shows a graph whose datapoint markers have been modified using the **Patterns** option and for which a color, alternate color, and pattern have been selected.
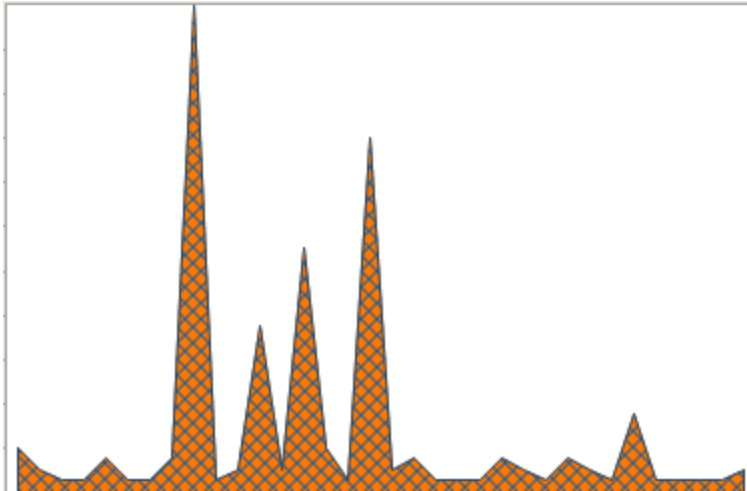
- **Gradients:** Displays the datapoint markers using shades of the colors that you select in the **Color** and **Alternate** lists. For example, the following image shows a graph whose datapoint markers are displayed using the **Gradients**, **Color**, and **Alternate** options.
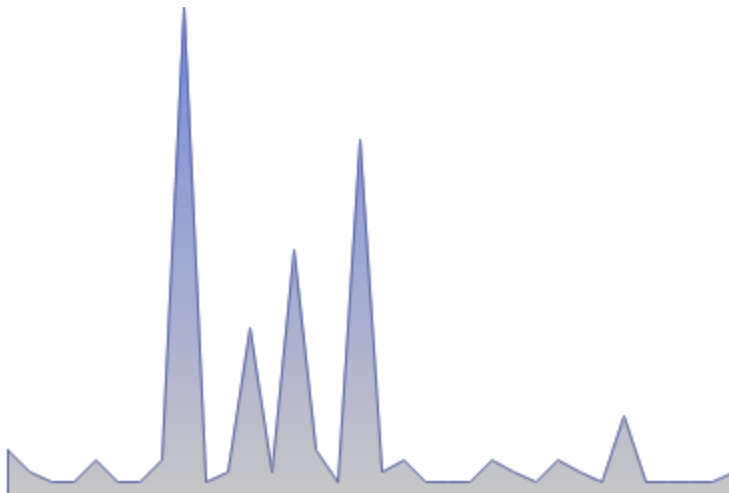
- **Color:** Contains a list of colors that you can apply to graph objects (e.g., bars, bubbles, and candlesticks). This list is enabled only if you select **Solid**, **Patterns**, or **Gradients** in the **Scheme** list.
- **Alternate:** Contains a list of colors that you can apply to graph objects. This list is enabled only if the **Patterns** or **Gradients** option is selected in the **Scheme** list. If you select a color in the **Alternate** list, it will be applied to the graph with the color that you selected in the **Color** list.

- **Pattern:** Contains a list of patterns that you can apply to graph objects. This list is enabled only if you selected **Monochrome** or **Patterns** in the **Scheme** list.

The Bar/Marker Section

The **Bar/Marker** section on the **Series** tab contains options that let you view and modify the way the datapoint markers are displayed. The options in this section vary depending on the style of the graph that you are viewing. If you are viewing a graph that does not display datapoint markers (e.g., pie chart), this section is empty.

The Barsection contains the following options:

- **Gap width:** A list that lets you specify the amount of space that appears between the datapoint markers in increments of 10. For example, the following image shows a bar graph whose gap width is *0*.
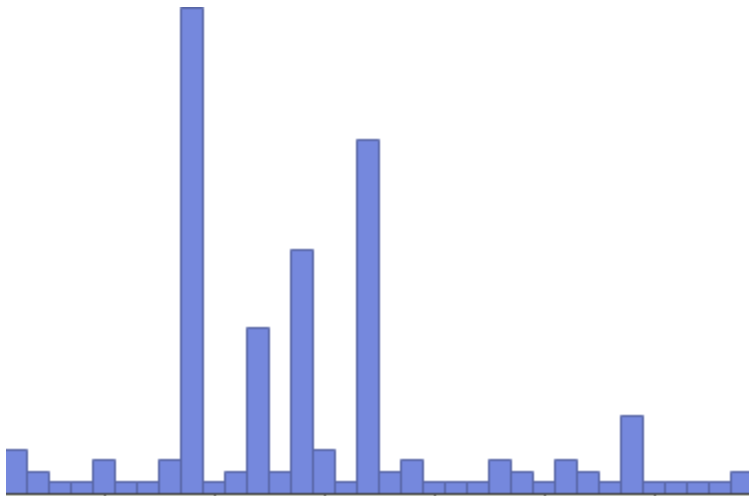
The following image shows the same graph, where the gap width has been changed to *50*.
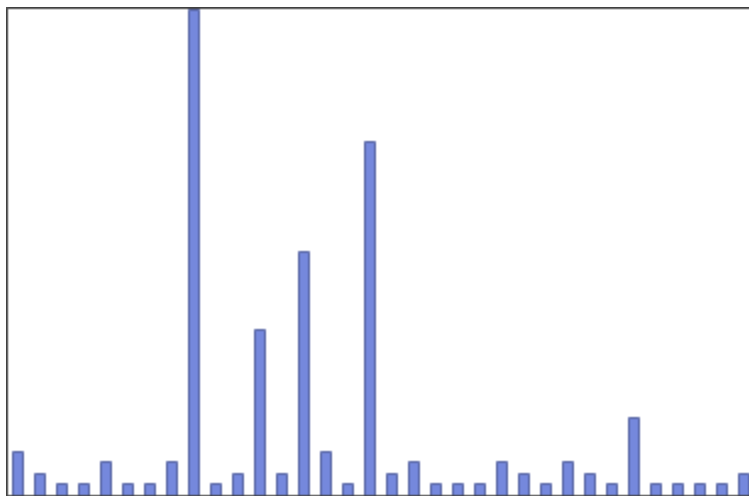
Note the space between the bars.

- **Square:** A radio button that you can select to display the datapoint markers in a square (or rectangular) shape. This is the default setting. For an example, see the previous image.

- **Cylinder:** A radio button that you can select to display the datapoint markers in a cylindrical shape. For example, the following image shows a bar graph displayed using the **Cylinder** option.
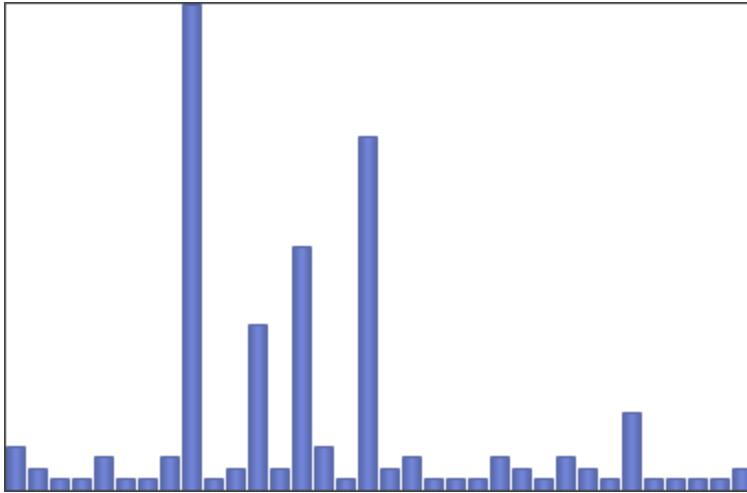
Note that the bars are rounded like cylinders.

- **Cone:** A radio button that you can select to display the datapoint markers in a cone shape. For example, the following image shows a bar graph displayed using the **Cone** option.
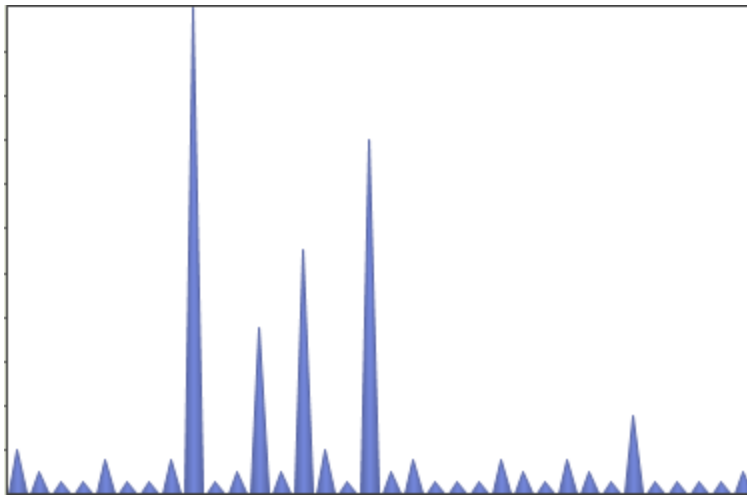
Note the pointed tips and rounded bottoms on the datapoint markers.

The **Markers**section contains the following options:

- **Shape:** A list that lets you specify the shape of the datapoint markers. For example, the following image shows a scatter graph whose marker shape has been modified to display triangles.
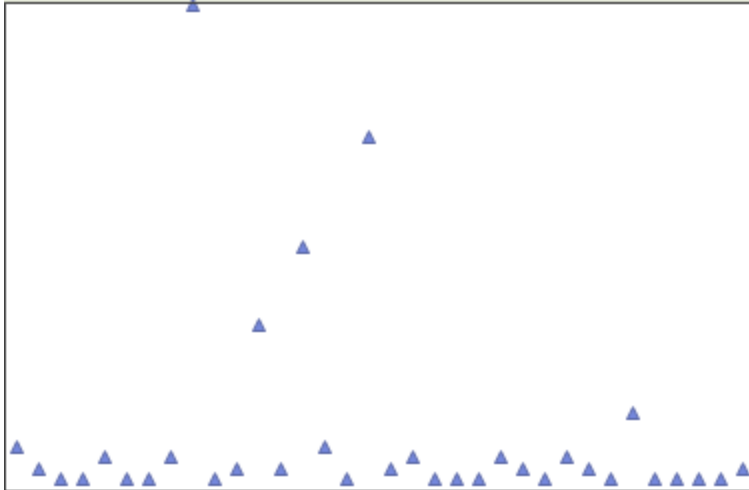
Note: The default shape for datapoint markers is *square.*

- **Size:** Lets you specify the size of the datapoint markers. For example, the following image shows the same graph that is shown in the preceding image, where the size of the datapoint markers has been set to 10.
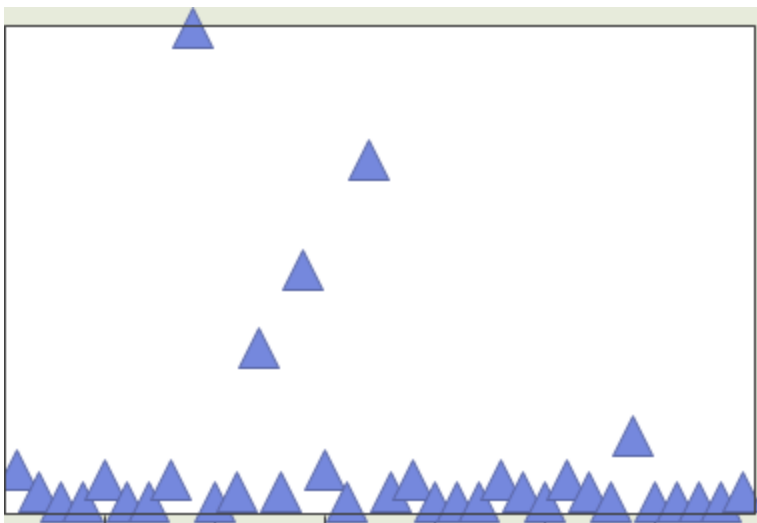
Note: The default size for a datapoint marker is *3.*

- **Same color lines:** A check box that you can select to display the lines that connect the datapoint markers in the same color as the markers themselves (e.g., as on a line graph). Note that this option is selected by default. For example, the following image shows a line graph whose lines are a the same color as the datapoint markers.
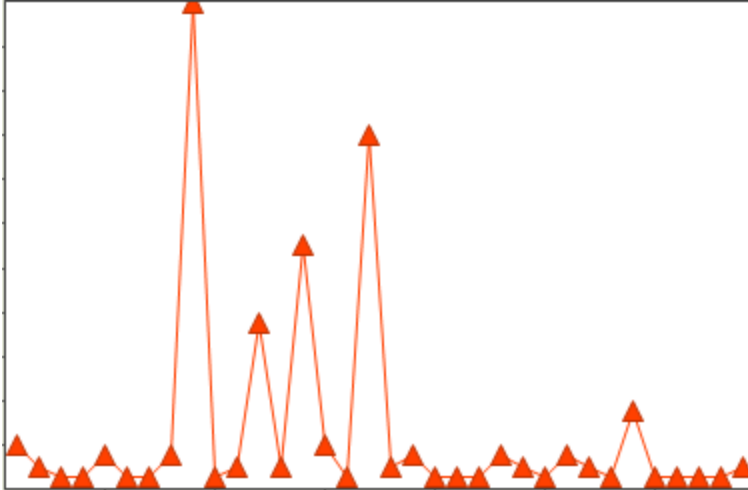
Note: Clearing the **Same color lines** check box will not change the color of the lines.

The Border Section

The **Border** section on the **Series** tab contains options let you define the way that the border around the datapoint markers appears. Note that this section is enabled only if the **Border** check box is selected, indicating that you want to modify the border on the datapoint markers. The following options are available:

- **Effect:** Lets you specify the style of border that you want to display around the datapoint markers. Note that the default setting is **Dark**. You can choose from the following options:
    - **None:** Displays the outline around each datapoint marker in the color and style tha~~t~~ ple, the following image sh~~o~~ ng the **None** option and wh~~ich~~ nodified.
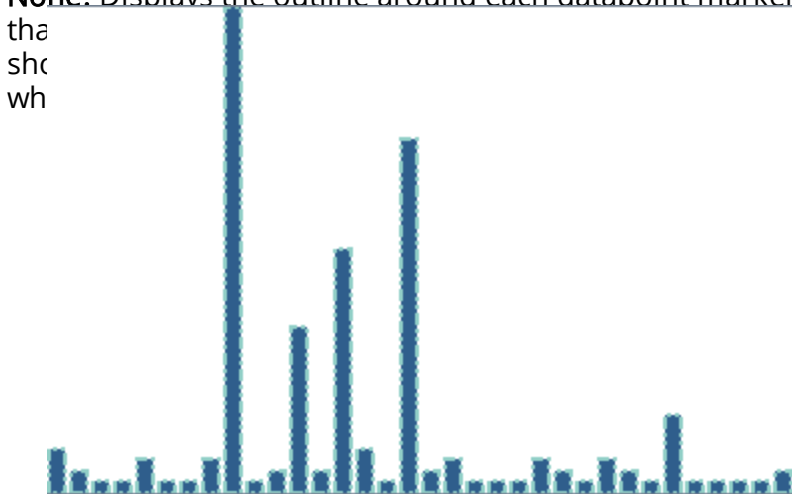
    - **Raised:** Displays the border around the datapoint markers using the options that appears in the **Style** and **Weight** lists. The **Raised** option looks similar to a 3D effect on the datapoint markers. For example, the following image shows a graph whose datapoint markers are displayed using a raised border.
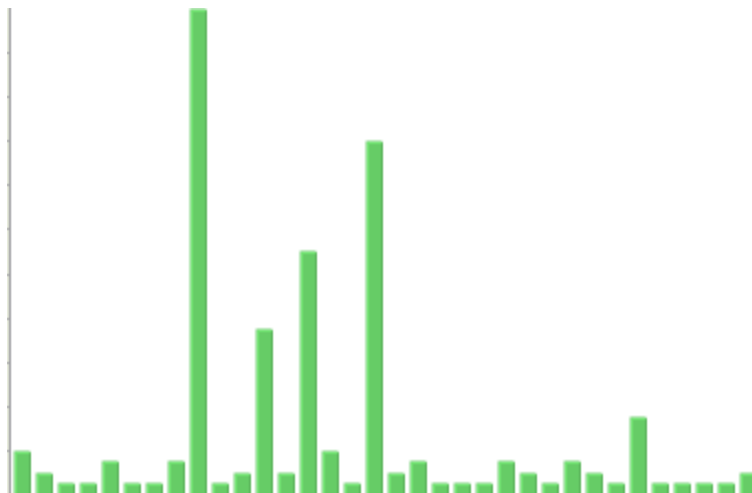
    - **Light:** Displays the border around the datapoint markers using the options that appear in the **Style** and **Weight** lists a shade lighter than the color that is applied to the datapoint markers themselves. For example, the following image shows a

bar graph whose datapoint markers are displayed using the **Light** border option. Additionally, the fill on the datapoint markers has been modified using the **Gradient** option to make the border more distinguishable.
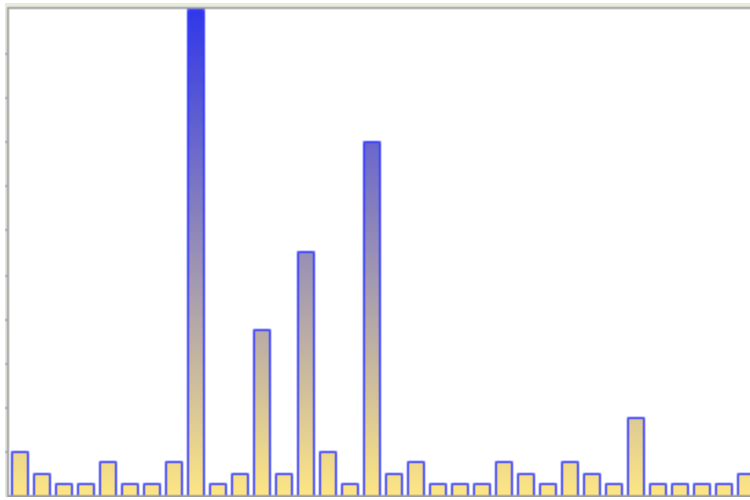
- **Dark:** Displays the border around the datapoint markers using the options that appear in the **Style** and **Weight** lists a shade darker than the color that is applied to the datapoint markers themselves. For example, the following image shows a bar graph whose datapoint markers are displayed using the **Dark** border option.
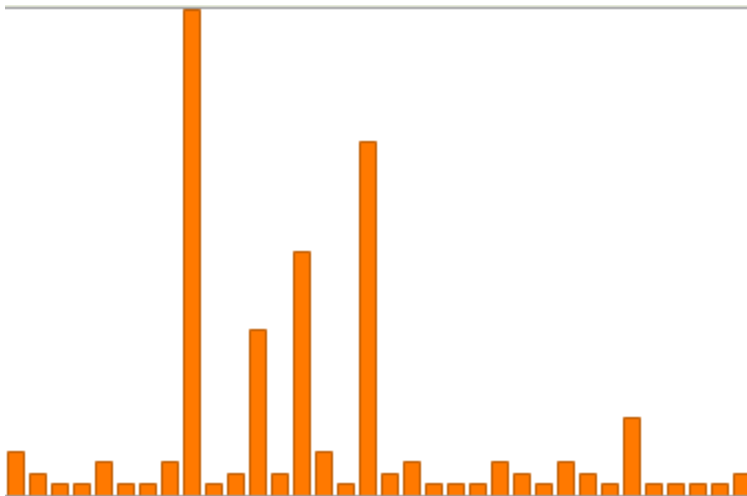
- **Opposite:** Displays the border around the datapoint markers using the options that appear in the **Style** and **Weight** lists in the same color that is applied to the series markers themselves. For example, the following image shows a bar graph whose series markers are displayed using the **Opposite** option and whose line style and thickness has been modified.
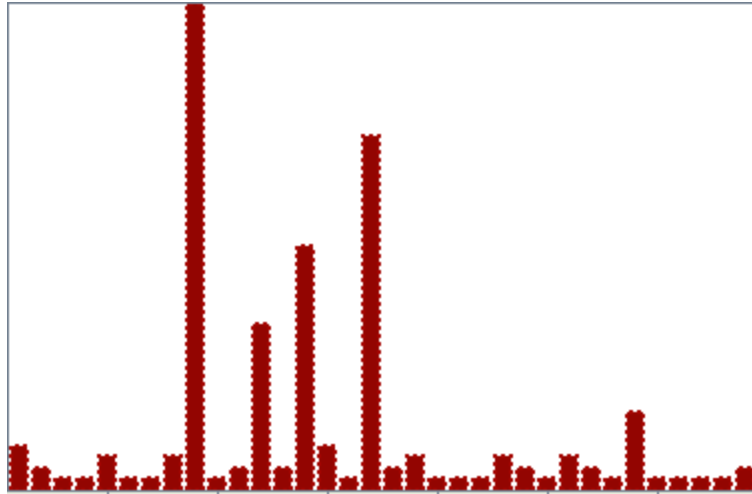
- **Shadow:** Displays a shadow behind and to the right of the series markers. For example, the following image shows a bar graph whose series markers are displayed using the **Shadow** option.
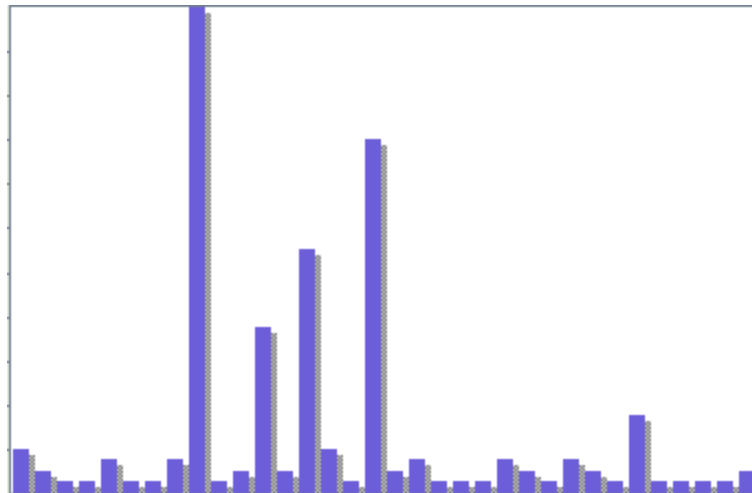
- **Color:** Lets you specify the color of the border lines. Note that this option is enabled only after you select the **None** option.
- **Style:** Lets you specify a style for the border lines. Note that this option is enabled *unless* the **Shadow** option is selected.
- **Weight:** A list that lets you specify the thickness of the border line. Note that this option is enabled *unless* the **Shadow** option is selected.

The Scale Section

The **Scale** section on the **Axis** tab contains options that let you specify the way that the scale is displayed on the axis. The following options are available in the **Scale** section:

- **Step:** Lets you specify the increment between the tick marks on the axis. Note that the default setting is *Auto*.

- **Minimum:** Lets you specify the lowest value that appears on the axis. For example, in a financial chart, your first value might be $0. If you are plotting a graph for high-priced equipment, you may set the minimum at $250,000. Note that the default settings is *Auto*.

- **Maximum:** Lets you specify the highest value that appears on the axis, much like the **Minimum** option. Note that the default option is *Auto*.

  > **Note:** The default value, *Auto*, indicates that the scale is determined automatically by the Meridium APM system based on the actual values plotted. The *Auto* setting ensures that the scale is large enough to accommodate all the values that you chose to plot. You can modify these values if desired.

- **Decimals:** Lets you specify the number of decimal places to which the values will be rounded when they are displayed on the axis.

- **Format:** Lets you specify what type of numbers appear on the axis and formats the scale accordingly. Your selection here does not take into account any format rules defined for the fields that have been plotted. You will need to choose a format that is appropriate for the values that you have selected to include in the graph.

  You can choose from the following options:

  - **None:** Displays the values as they appear in the underlying query or dataset.

  - **Number:** Displays the values as whole numbers, using the number of decimal places that you specify in the Decimals list.

  - **Currency:** Displays the values as currency (e.g., using dollar signs).

  > **Note:** If you choose to format numeric values as currency, the Meridium APM system will use the currency symbol associated with the Windows Regional Options on the Meridium APM Framework workstation where the graph is being viewed. In graphs that will be viewed by different users at different workstations, this currency symbol may or may not be appropriate for the values that are actually displayed on the graph.

- **Scientific:** Displays the values using a format that is useful for shortening larger numbers to a more manageable size. For example, if the value *120,000* appears on the axis, after the **Scientific** option is applied to the graph, this number will appear as *1.2E+005*, which is the same as saying 1.2 x 100,000.

- **Percentage:** Multiplies the values by 100 and then displays the calculated values as percentages, using percent signs. Note that this option assumes that the

underlying values are already converted to decimals such that multiplying them by 100 will yield the expected result. For example, if the underlying value is .25, selecting the **Percentage** option will multiple .25 by 100, yielding 25, and then format 25 with a percent sign. So, the graph will display *25%*.

- **Logarithmic:** Lets you specify that you want to display values as logarithms. This option is useful when you are working with a wide range of values so that the graph can display the values on a more manageable scale. For example, the following image shows a graph whose values on the scale are displayed using the **Number** format option and the **Logarithmic** check box is cleared.
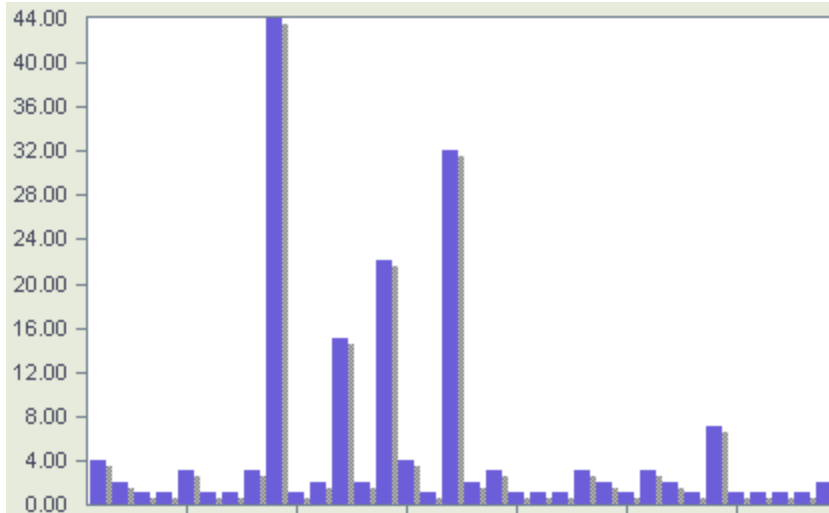
If you were to *select* the **Logarithmic** check box, the same graph would appear as shown in the following image.
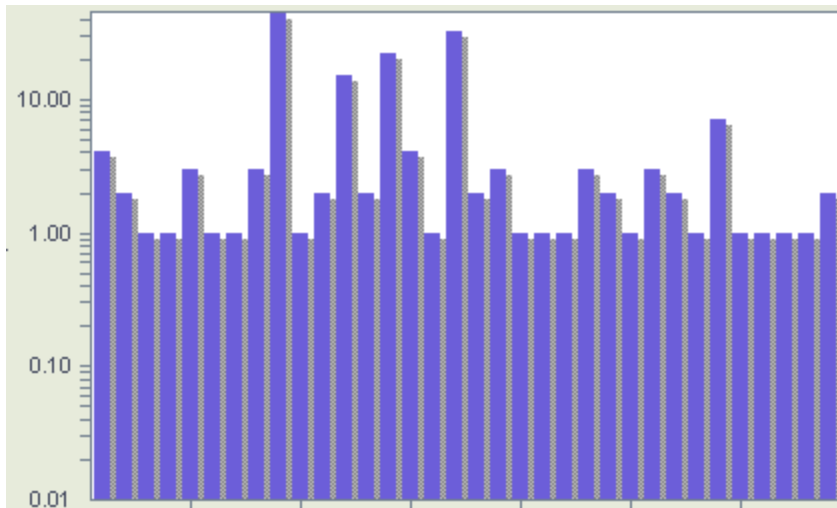
The Labels Section

The **Labels** section on the **Axis** tab contains options that let you specify the way that labels are displayed on the axis. To modify the axis label of a different axis, you must access the **Properties** dialog box using the shortcut menu on that axis. For example, if you want to modify the axis labels on the x-axis, right-click the x-axis labels, and then click **Properties**.

The following options are available in the **Labels** section:

- **Angle:** Lets you specify the direction in which the labels will be displayed on the axis. You can choose from the following options:
    - **Vertical:** Displays the labels vertically on the axis.
    - **Horizontal:** Displays the labels horizontally on the axis.
    - **45 Degrees:** Displays the labels at a 45 degree angle on the axis.
- **Staggered:** A check box that you can select to display axis values in a staggered formation to prevent them from overlapping.

The Gridlines/Tickmarks Section

The **Gridlines/Tickmarks** section on the **Axis** tab contains options that let you specify the way the gridlines and tick marks are displayed on the axes. Note that to modify the gridlines or tick marks on a different axis, you must access the **Properties** dialog box via the shortcut menu *on the axis you want to modify*. For example, if you want to modify the gridlines or tick marks on the x-axis, right-click the x-axis, and click **Properties**.

The following options are available in the **Gridlines/Tickmarks** section:

- **Show Gridlines:** A check box that you can select to display the gridlines on a graph. Gridlines are lines that span the length or width of the graph at defined intervals to provide you with a point of reference. You can modify the color and line style of the gridlines, if desired. For example, a graph whose gridlines are displayed on the x-axis using the color that appears in the **Color** list would look similar to this:
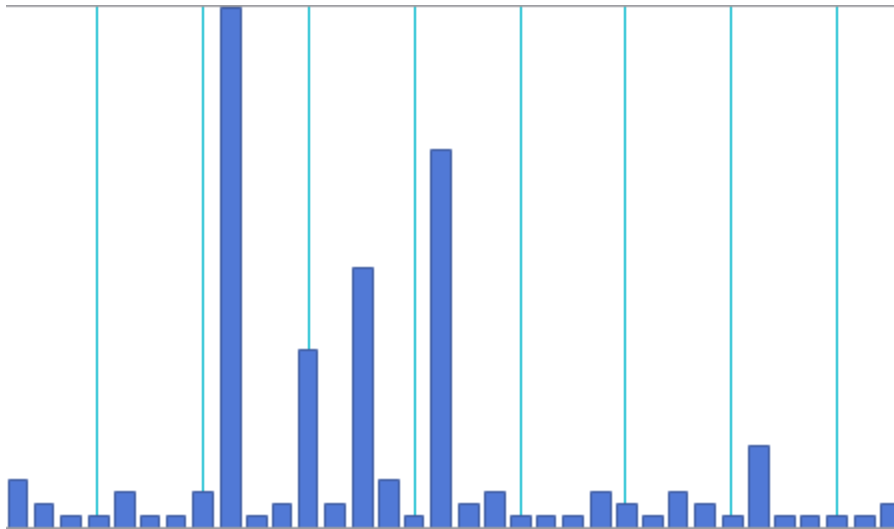
...and a graph whose gridlines are displayed on both the x- and y- axes using the color and style that appear in the **Color**, **Style**, and **Weight** lists would look similar to this:
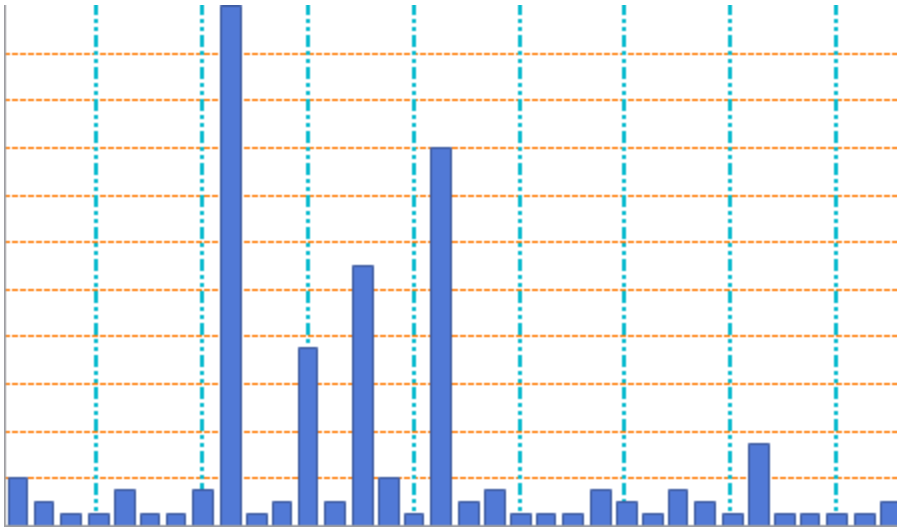
- **Tickmark:** Lets you specify where the tick marks for the axis values should appear. The tick marks are the short lines along the frame that indicate a value on the axis. You can choose from the following:

  - **None:** Hides the tick marks. For example, the following image shows a graph whose tick marks are not displayed.
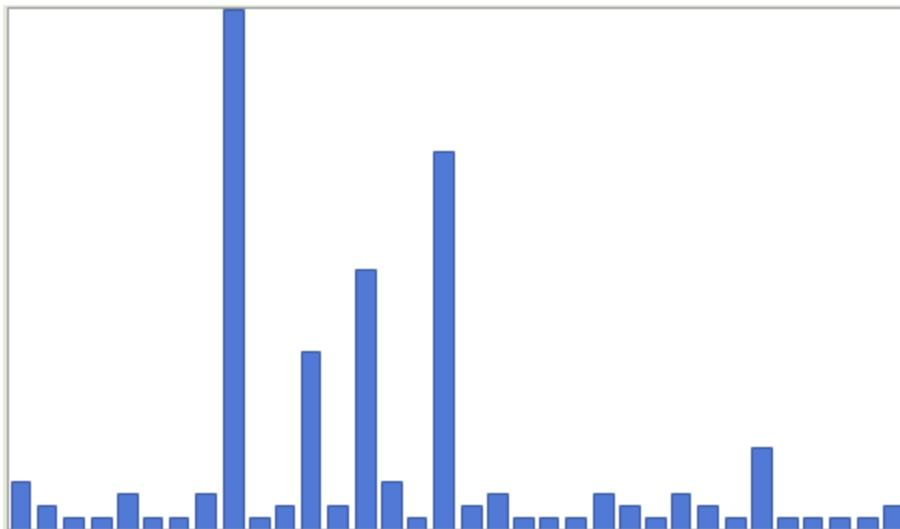
  - **Outside:** Displays the tick marks from the outside edge of the frame to the value on the axis. You can modify the color and the style of the tick marks if desired. For example, the following image shows a graph whose y-axis tick marks are displayed using the **Outside** option.
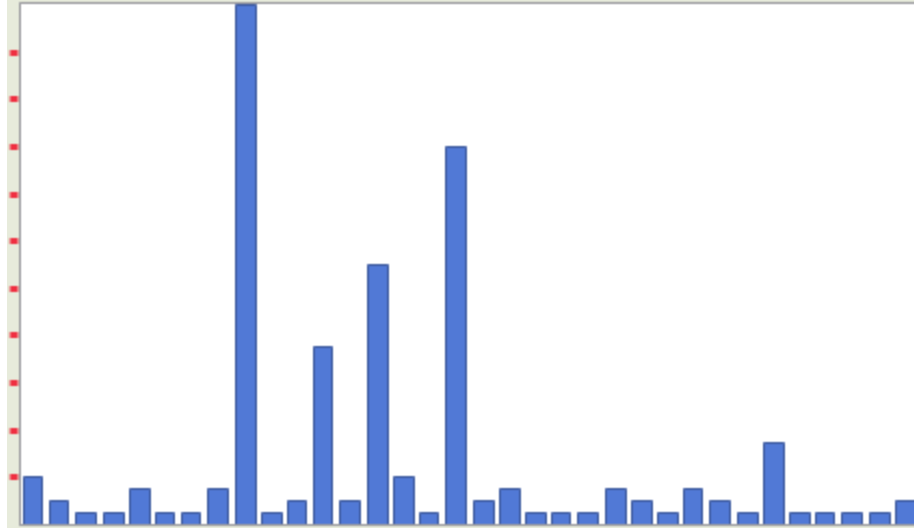
- **Inside:** Displays the tick marks within the frame of the graph (in the chart box area). You can modify the color and line style of the tick marks if desired. For example, the following image shows a graph whose y-axis tick marks are displayed using the **Inside** option.
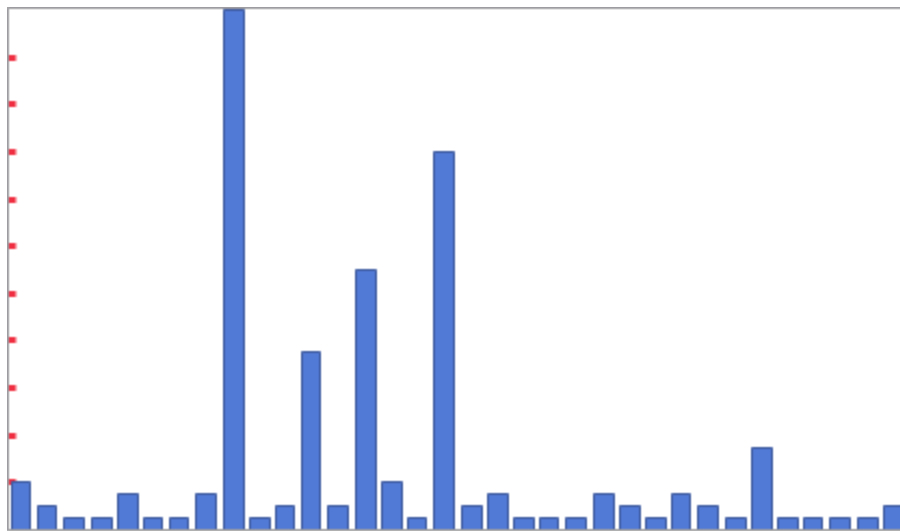
- **Across:** Displays the tick marks from inside the chart box to the value on the axis across the outer edge of the frame. You can modify the color and line style of the tick marks if desired. For example, the following image shows a graph whose tick marks are displayed using the **Across** option.
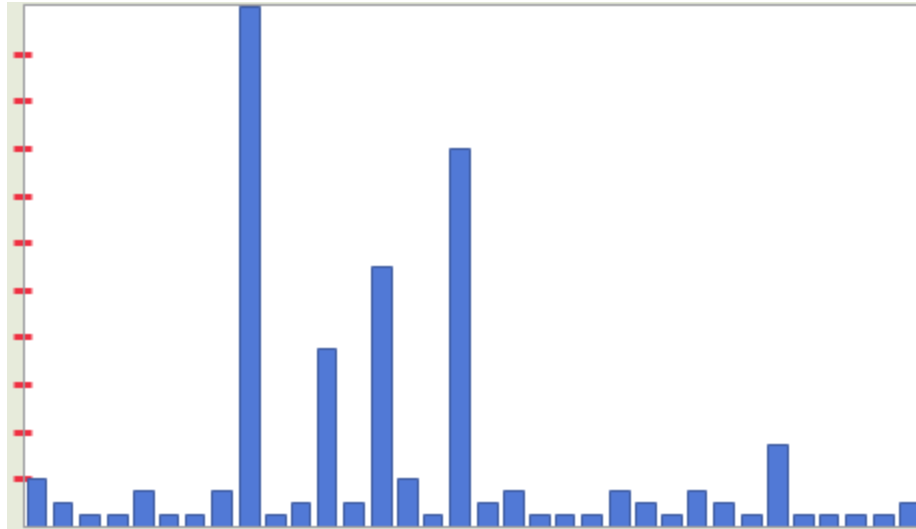
- **Color:** Lets you specify the color of the gridlines and tick marks.
- **Style:** Lets you specify the way the lines will appear in the gridlines and tick marks.
- **Weight:** Lets you specify the thickness of the gridlines and tick marks.
- **Interlaced:** A check box that you can select to displays the space between the gridlines. You can modify the color of the interlaced gridlines if desired. Note that to display the interlaced gridlines, the **Show Gridlines** option does not need to be selected. For example, the following image shows a graph whose interlaced gridlines are displayed and whose gridlines are not.
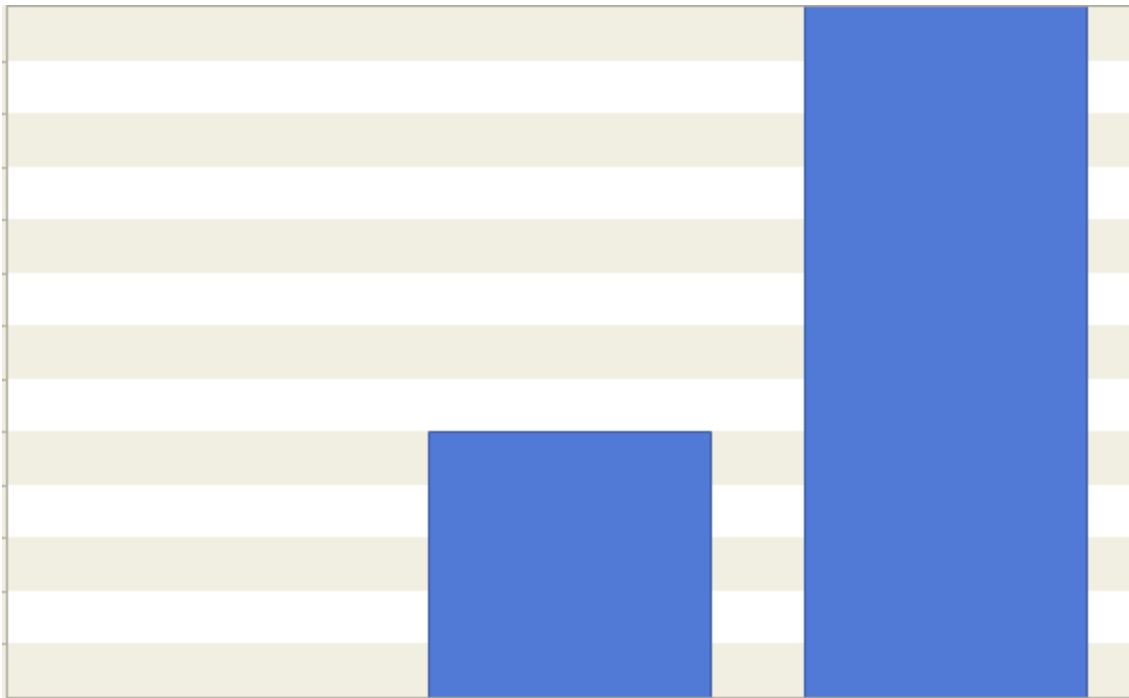
- **Color:** When the **Interlaced** option is selected, this list lets you specify the color of the interlaced gridlines.

Modifying Graph Colors

If desired, you can modify the appearance of a graph by changing the colors that appear in various locations in a graph. You can change ALL colors on a graph by modifying the color palette, which applies a predefined palette of colors to the entire graph. Or, you can modify the color of each area on the graph individually.

The following instructions provide details on applying a color palette to the entire graph and on modifying the colors in specific areas of the graph.

To apply a color Palette to the entire graph:

1. In the Meridium APM Framework, on the **Graph Viewer** page, on the standard tool-bar, click the **Palette Selector** button.

   A drop-down list appears, displaying the palette options.

2. Click the desired color palette.

   The colors that appear on the palette are applied to the graph. You can further modify the colors on the graph, as desired.

3. On the **Common Tasks** menu, click the **Save Graph** link.

   Your changes are saved. The next time you view the graph, you will see your changes.

To modify the color of the background, chart box, or series area on the graph:

1. Right-click anywhere inside the area whose colors you want to modify.

   A shortcut menu appears.

2. On the shortcut menu, point to **Color**.

   A submenu appears, displaying a color chart.
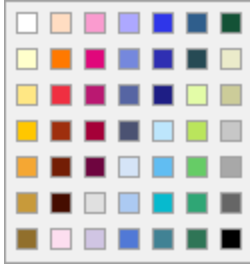
3. Click the desired color.

    The area is displayed using the selected color.

4. On the **Common Tasks** menu, click the **Save Graph** link.

    Your changes are saved. The next time that you view the graph, you will see your changes.

Modifying the Appearance of X-Axis Labels

If desired, you can modify the way that the labels are displayed on x-axis the graph. The following instructions provide details on modifying the appearance of the x-axis labels.

To modify the appearance of x-axis labels on the graph:

1. In the Meridium APM Framework, on the **Graph Viewer** page, on the standard toolbar, click the **Axes Settings** button.

   A submenu appears.

2. In the **X Axis** section, click the option that you want to apply to the graph. You can choose from the following options:

   - **Vertical Labels:** Displays the labels veritcally on the x-axis.
   - **Staggered:** Displays the labels on the x-axis in a staggered formation to prevent overlay.
   - **Show Labels:** Displays the labels on the x-axis.

   > **Note:** A ☑ icon to the left of an option indicates that the corresponding setting is currently applied to the graph.

3. On the **Common Tasks** menu, click the **Save Graph** link.

   Your changes are saved. The next time that you view the graph, you will see your changes.

Displaying Gridlines

If desired, you can display the gridlines and the space between the gridlines on the graph. The following instructions provide details on displaying the gridlines and interlaced shading between the gridlines.

Note that after you have applied the desired modifications, you should click the **Save Graph** link on the **Common Tasks** menu so that the next time you view the graph, you will see your modifications.

To display grid lines in a graph:

1. In the Meridium APM Framework, on the **Graph Viewer** page, on the standard toolbar, click the **Axes Settings** button.

   A submenu appears.

   The submenu contains the **Y Axis** section and the **X Axis** section and includes the additional option, **Options**, which displays the **Properties** dialog box, where you can view and modify the current settings for the axes.

2. In the **Y Axis** section, click the option that you want to apply to the graph. You can choose from the following options:

   - **Grid:** Displays the gridlines along the y-axis. An outline will appear around the icon to the left of the **Grid** option indicating that the setting is currently applied to the graph.

   - **Interlaced:** Displays the space between gridlines along the y-axis. A icon to the left of an option indicates that the corresponding setting is currently applied to the graph.

   Your changes are applied to the graph.

   For example, in the following image, you can see that both the **Grid** and **Interlaced** options have been applied to the graph. This means that on the graph, the gridlines and the spacing between the lines are being displayed on the graph.
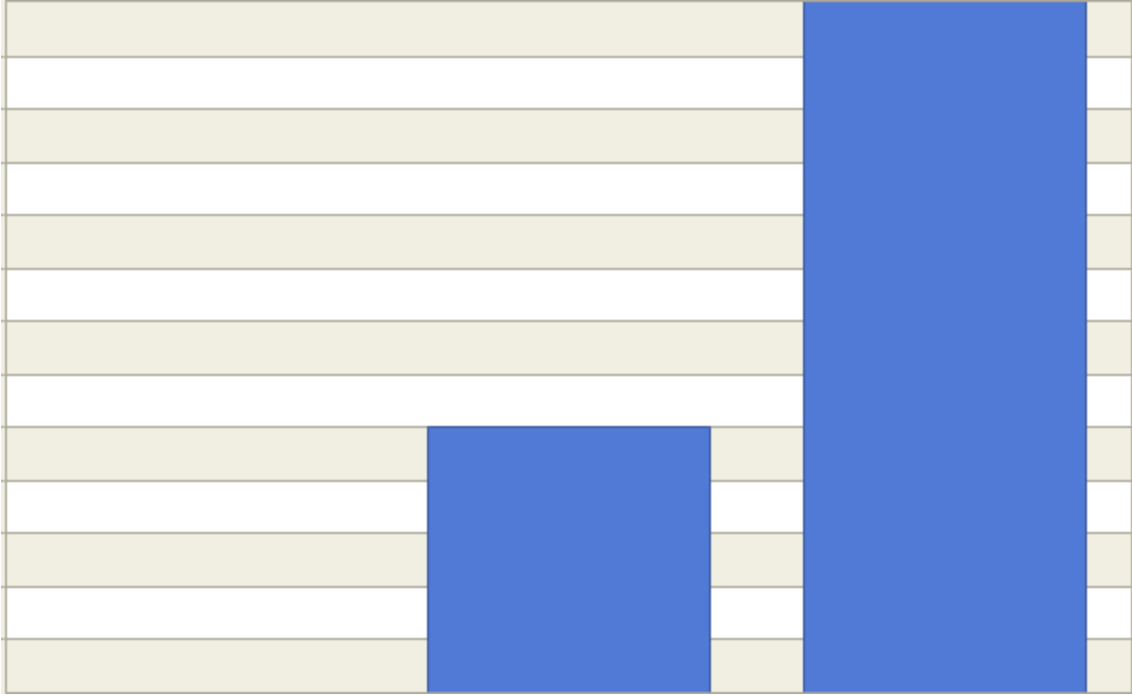
Modifying Graph Fill Options

If desired, you can modify the fill colors and patterns that are applied to the graph. For example, the following image shows a bar graph whose bars have a specific fill color and pattern.
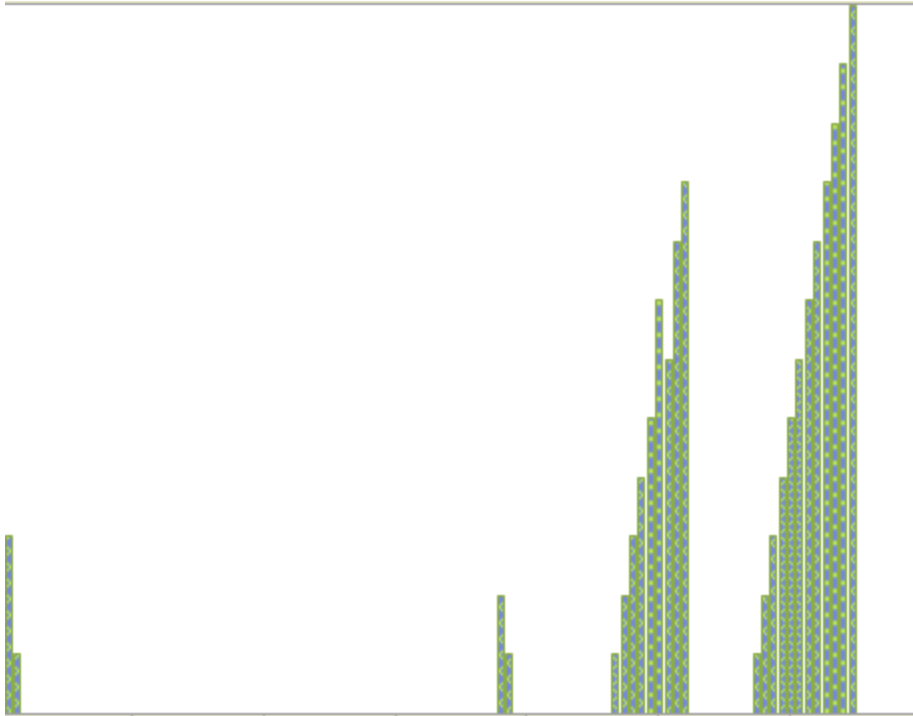
To modify the fill options applied to a graph:

1. In the Meridium APM Framework, on the **Graph Viewer** page, right-click the graph, and on the shortcut menu, click **Properties**.

   The **Properties** dialog box appears.
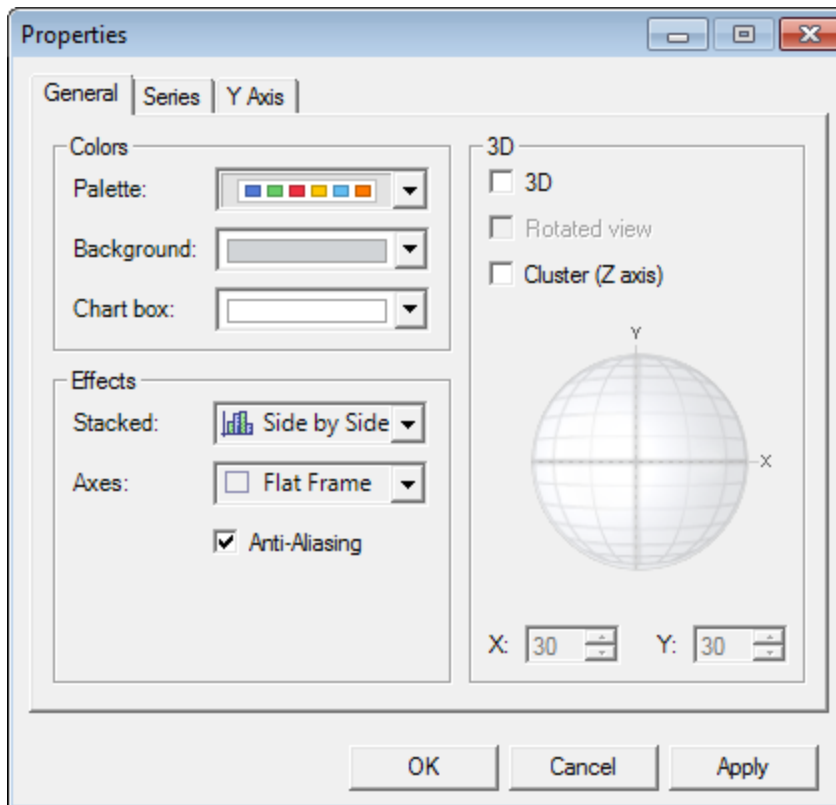
2. Click the **Series** tab.

   The **Series** tab appears, displaying the current settings that are applied to the graph.
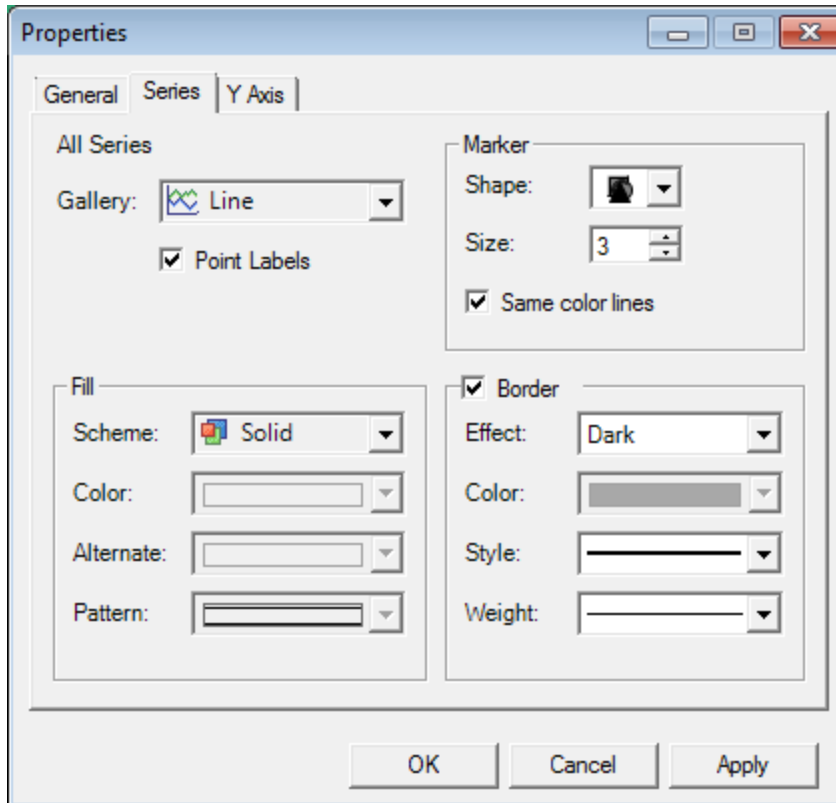
3. In the **Fill** section, select the desired settings. Note that some of the options may be disabled based on other settings that are applied to the graph. For example, if in the **Scheme** list, you select **Solid**, then the **Alternate** and **Pattern** options will be disabled.

4. Click the **Apply** button.

   The settings are applied to the graph.

   At this point, you can continue modifying graph settings or click **OK** to close the **Properties** dialog box and return to the graph.

5. On the **Common Tasks** menu, click the **Save Graph** link.

   Your changes are saved. The next time that you view the graph, you will see your changes.

Modifying the Appearance of Lines in a Graph

If desired, you can modify the appearance of the lines that outline an item (e.g., a bar) in a graph. For example, the following image shows a graph using the default outline settings.
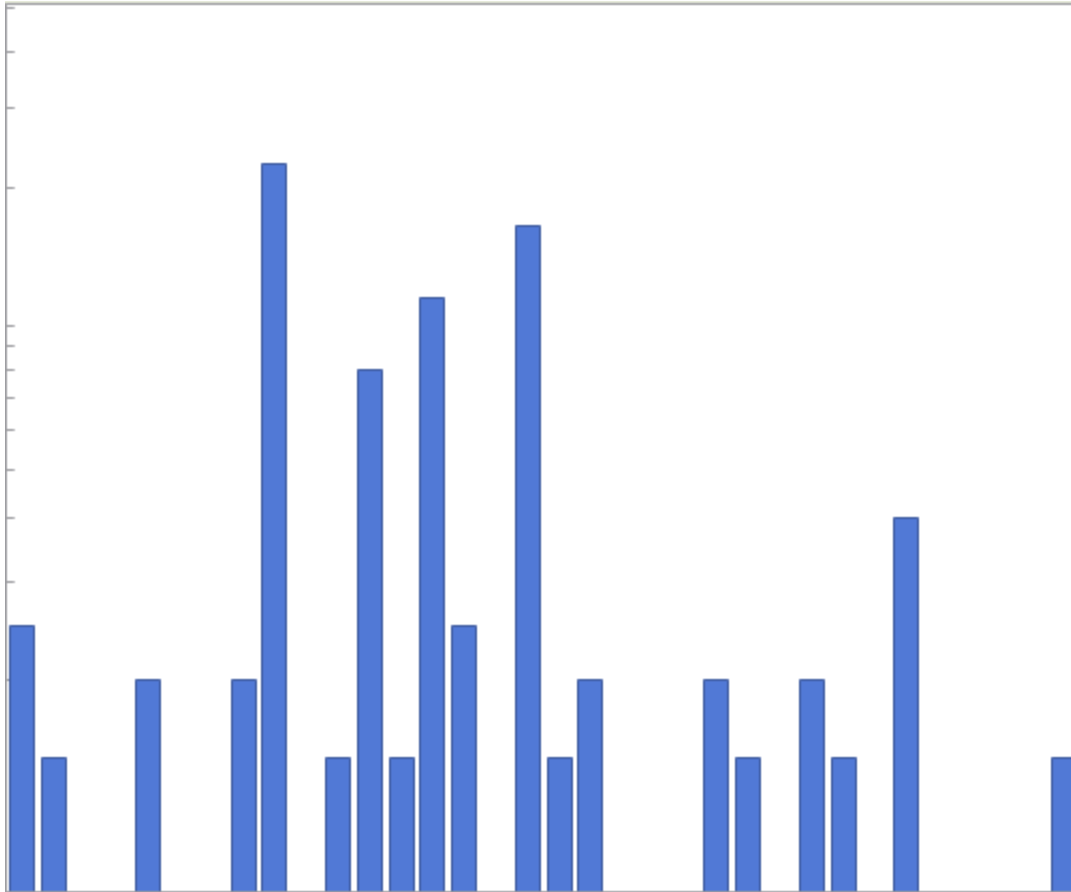
The following image shows the same graph after the line style and weight have been modified via the **Properties** dialog box.
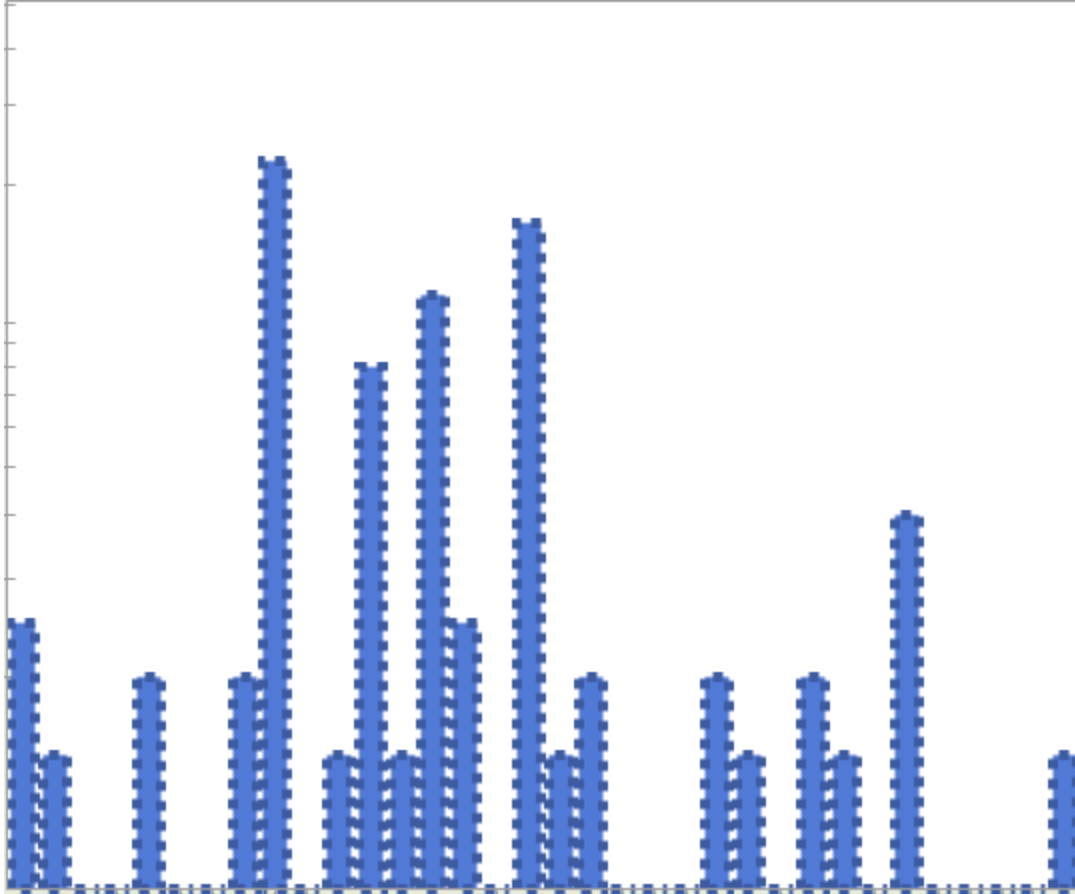
To modify the lines that appear on the series in a graph:

1. In the Meridium APM Framework, on the **Graph Viewer** page, right-click the series area on the graph, and on the submenu, click **Properties.**

    The **Properties** dialog box appears, displaying the **Series** tab, which contains the current settings that are applied to the series on the graph.
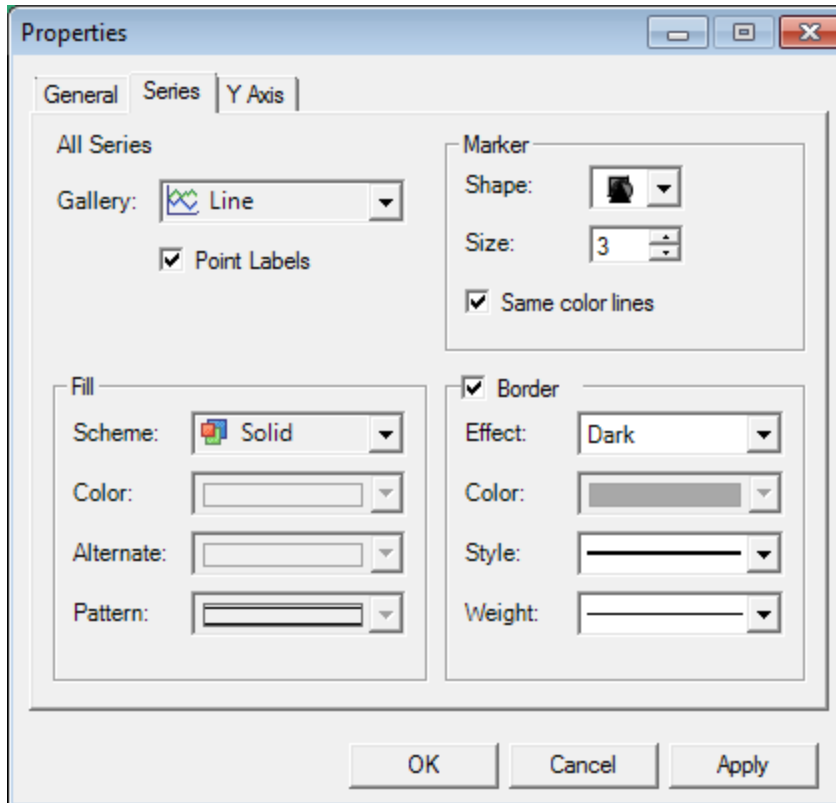
2. In the **Border** area, be sure that the **Border** check box is selected.

   The options in the **Border** area are enabled.

3. You can choose from the following options:

- **Effect:** A list that lets you specify the style of border that you want to display. This list contains the following options:
  - **None**
  - **Raised**
  - **Light**
  - **Dark**
  - **Opposite**
  - **Shadow**
- **Color:** A list that lets you specify the color of the border.
- **Style:** A list that lets you specify the line style that you want to display as the border.
- **Weight:** A list that lets you specify the thickness of the border line.

4. Click the **Apply** button to apply your changes to the graph and keep the **Properties** dialog box open, or click **OK** to close the **Properties** dialog box and return to the graph.

5. On the **Common Tasks** menu, click the **Save Graph** link.

   Your changes are saved. The next time that you view the graph, you will see your changes.

Adding Text Box or Balloon to a Graph

On the **Graph Viewer** page, you can add a text box or a balloon to the graph.

To add a text box or balloon to a graph:

1. In the Meridium APM Framework, on the **Graph Viewer** page, on the annotation toolbar, click the **Text** button or **Balloon** button 💬 .

   The pointer appears as a plus sign.

2. Click and drag the pointer to the desired location, creating a box.

   The text box or balloon appears. The default text inside the text box or balloon is **Text**. For example, the following image shows a graph with a text box and a balloon. The text box is the rectangle, and the balloon is the rectangle with rounded edges and a triangular extension.
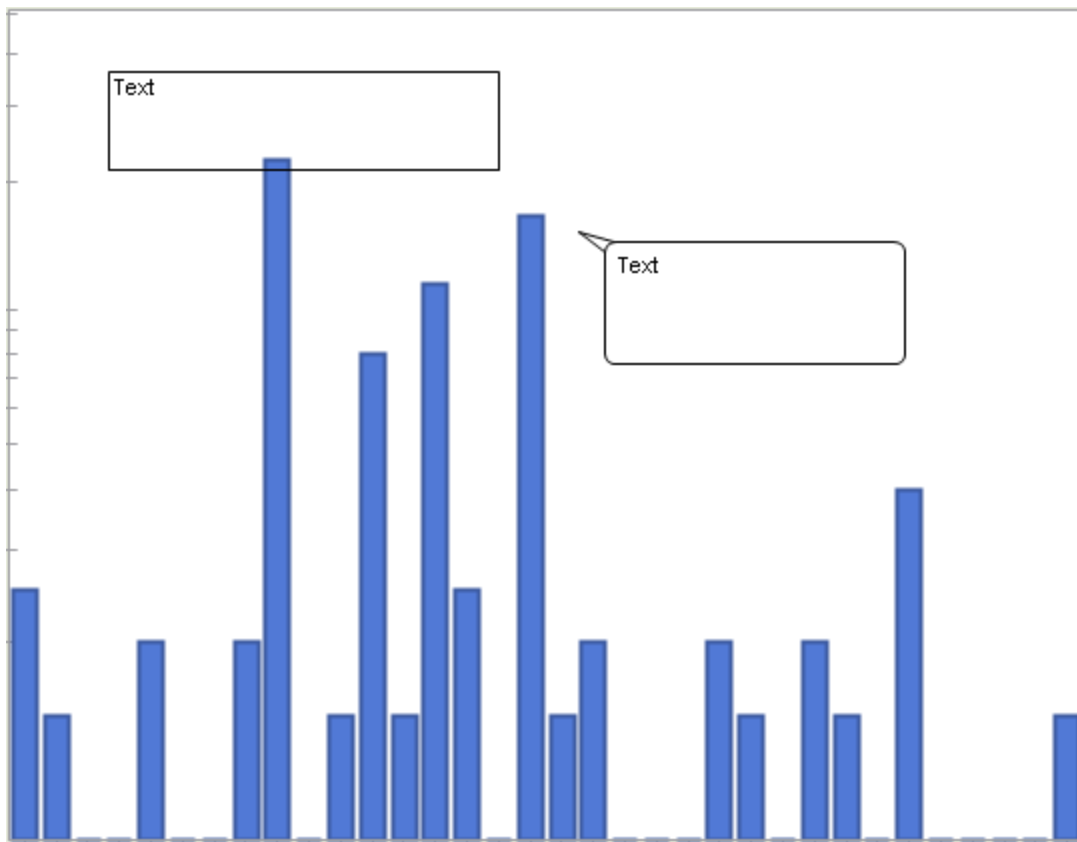
3. Replace the default text with the desired text.

   If desired, you can move, re-size, modify the outline and fill color of the text box or balloon, or delete the text box or balloon.

4. On the **Common Tasks** menu, click the **Save Graph** link.

Your changes are saved. The next time that you view the graph, you will see your changes.

To move a text box or balloon:

1. Pause on the outline of the text box until the pointer appears as a four-way arrow.
   ✥

2. Click and drag it to the desired location.

   The text box or balloon appears in the selected location.

To remove a text box or balloon:

1. Pause on the outline of the text box until the pointer appears as a four-way arrow.
   ✥

2. Right-click the text box or balloon, and on the shortcut menu, click **Delete**.

   The text box or balloon is removed from the graph.

Inserting an Image into a Graph

On the **Graph Viewer** page, you can insert an image into a graph.

To insert an image into a graph:

1. In the Meridium APM Framework, on the **Graph Viewer page**, on the annotation toolbar, click the **Picture** button 🖼 .

2. In the graph, select the region where you want to insert the image. To select a region, click a location, then and drag the pointer to create a box.

   An empty square appears, representing the area where the image will be placed.

3. Right-click inside the square.

   A shortcut menu appears.

4. On the shortcut menu, click **Image**.

   The **Open** dialog box appears allowing you to browse for the desired image.

5. Navigate to the location of the desired image, and select the image that you want to add to the graph.

6. Click **OK**.

The selected image appears in the graph. If desired, you can move the image, re-size the image box, modify the outline and fill color of the box surrounding the image, or delete the image.

7. On the **Common Tasks** menu, click the **Save Graph** link.

Your changes are saved. The next time that you view the graph, you will see your changes.

To move the image:

1. Point to the image.

The pointer appears as a four-way arrow. ✥

2. Click and drag the image to the desired location.

The image appears in the selected location.

To remove the image:

1. Point to the image.

   The pointer appears as a four-way arrow. ✛

2. Right-click the image, and on the shortcut menu, click **Delete**.

   The image is removed from the graph.

To re-size the text box surrounding the image:

1. Point to the edge of the text box surrounding the image.

   The pointer appears as a two-way arrow.

2. Click and drag the edge of the text box to the desired size.

   > **Note:** If you re-size the text box to a size larger than the image itself, the fill color of the text box will be white. You can [modify the fill color](#) if desired.

Modifying the Outline and Fill Color of Annotations

After you add an annotation to a graph, you can modify the outline and fill color of these annotations. For example, after you add a text box to a graph, you can change the color of the outline of the text box as well as the color inside the text box.

Note that annotations include text boxes, balloons, images, arrows, arcs, and shapes.

The following image shows a graph containing a text box whose outline and fill color have been modified.
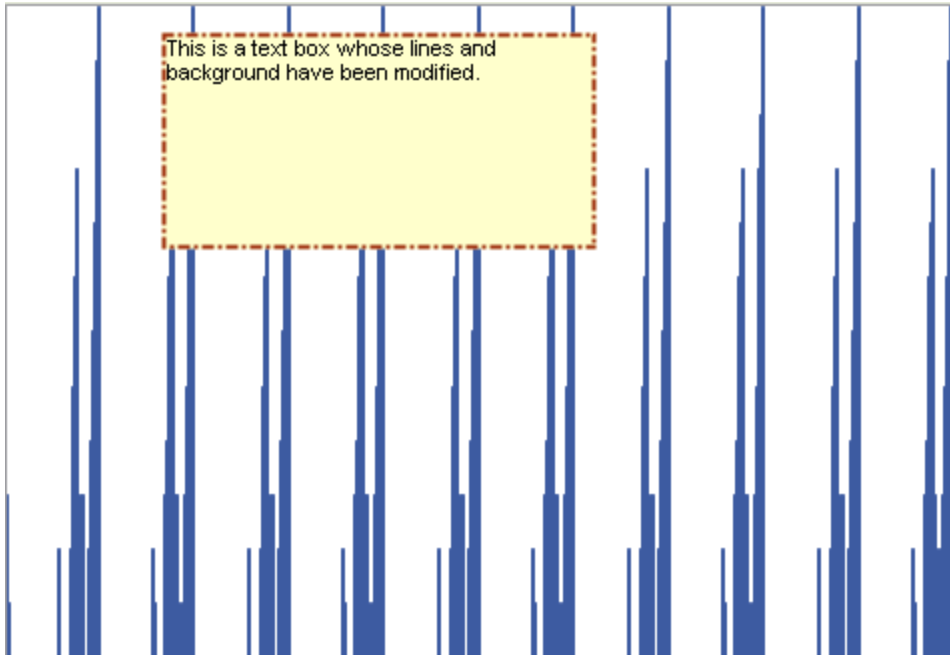
To modify the outline and fill color of annotations:

1. In the Meridium APM Framework, on the **Graph Viewer** page, right-click the annotation.

   A shortcut menu appears.

   > **Note:** For some annotations, to access the shortcut menu, you will need to point to the edge of the item until the pointer appears as a four-way arrow.

2. Point to the desired option, and select the desired option(s) from the submenu that appears.

   > **Note:** To see the fill color of images, you will need to re-size the text box surrounding the image beyond the size of the image itself.

3. Click outside the annotation.

   Your changes are reflected on the annotation.

4. On the **Common Tasks** menu, click the **Save Graph** link.

   Your changes are saved to the graph. The next time that you view the graph, you will see you changes.

Displaying the Underlying Values of Datapoints

The following instructions explain how to display the values associated with datapoints. You can display the values associated with ALL datapoints or with specific datapoints (depending on where you are viewing the graph).

To display the underlying values of datapoints:

1. In the Meridium APM Framework, on the **Graph Viewer** page:

    - If you want to display ALL datapoint values, right-click anywhere on the graph (other than on a specific datapoint).

    -or-

    - If you want to display a specific datapoint value, right-click the desired datapoint.

    The shortcut menu appears.

2. On the shortcut menu, click **Point Labels**.

    The value associated with the specified datapoints appears on the graph. For example, the following image shows a graph where only a specific datapoint value, *300*, is displayed.
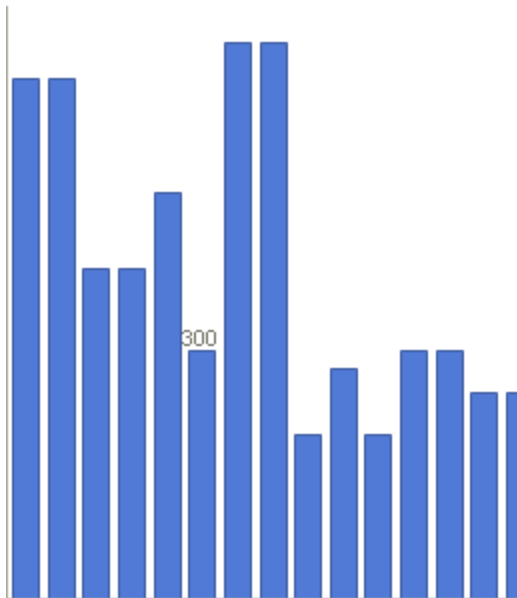
    Note that the values that appear are determined based on the settings that are defined in the **Graph Setup** window. You cannot determine by looking at the graph whether you are seeing y-axis values or some other values from the underlying query or dataset.

    > Œ**Hint:** To remove the values from the graph, right-click the graph again and click **Point Labels**.

3. On the **Common Tasks** menu, click the **Save Graph** link.

   Your changes are saved. The next time that you view the graph, you will see your changes.

Displaying a Trend Line on a Scatter Graph

If you are viewing a scatter graph, you can display a trend line whose slope and Y-intercept values are automatically calculated by the Meridium APM system.

To display a trend line on a scatter graph:

1. In the Meridium APM Framework, on the **Graph Viewer** page, on the scatter graph, right-click a datapoint, and then on the shortcut menu, click **Add Trend Line**.

   The trend line appears.
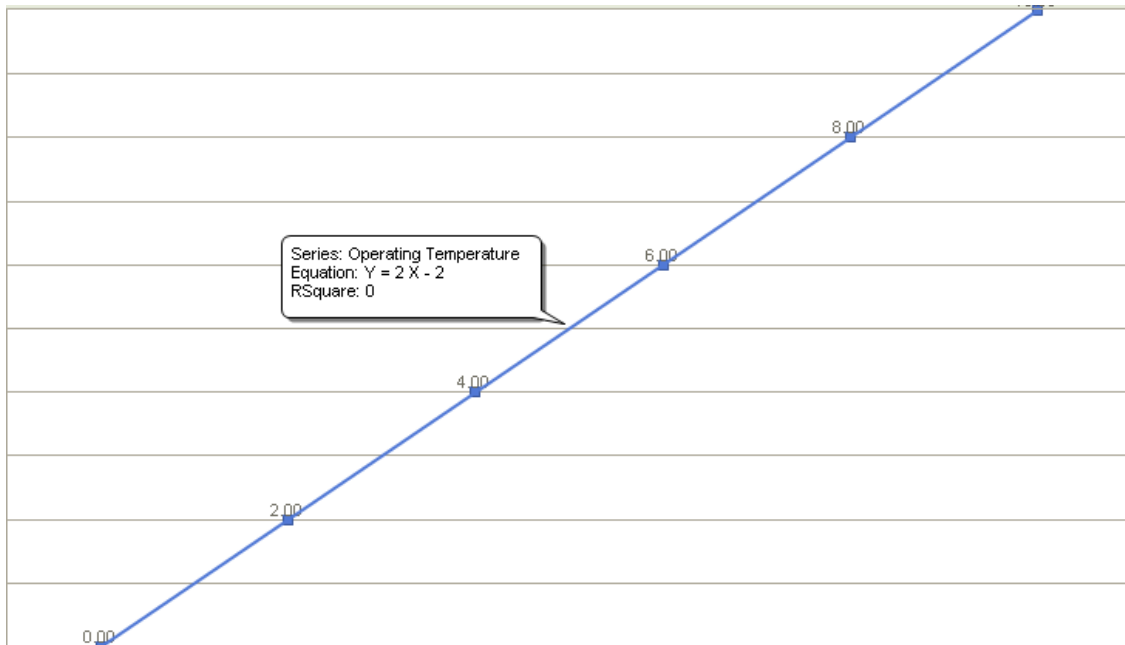
   In the annotation, you can see the equation of the line, where the slope and the Y-intercept value have been calculated by Meridium APM. The equation is presented as **y=mx+b**, where **m** is the slope and **b** is the Y-intercept.

   In this example, you can see that the slope of the line is **2**, and the point where the line crosses the y-axis is **-2**.

   You can also see the RSquare value of the line, which indicates how closely all datapoints lie to the trend line. In this example, the RSquare value is **0**.

   Œ**Hint:** To remove the trend line, select the line and press the Delete key.

2. On the **Common Tasks** menu, click the **Save Graph** link. The next time you open the graph, the trend line will be displayed.

Displaying the Data Editor

The Data Editor lets you view the underlying values associated with the graph you are currently viewing. Despite its name, however, you cannot actually modify data via the Data Editor.

To display the Data Editor:

- In the Meridium APM Framework, on the **Graph Viewer** page, right-click any blank space on the graph, and then on the shortcut menu, click **Data Editor**.

  -or-

- On the standard toolbar, click the **Data Editor** button 　.

  The values appear at the bottom of the graph window.
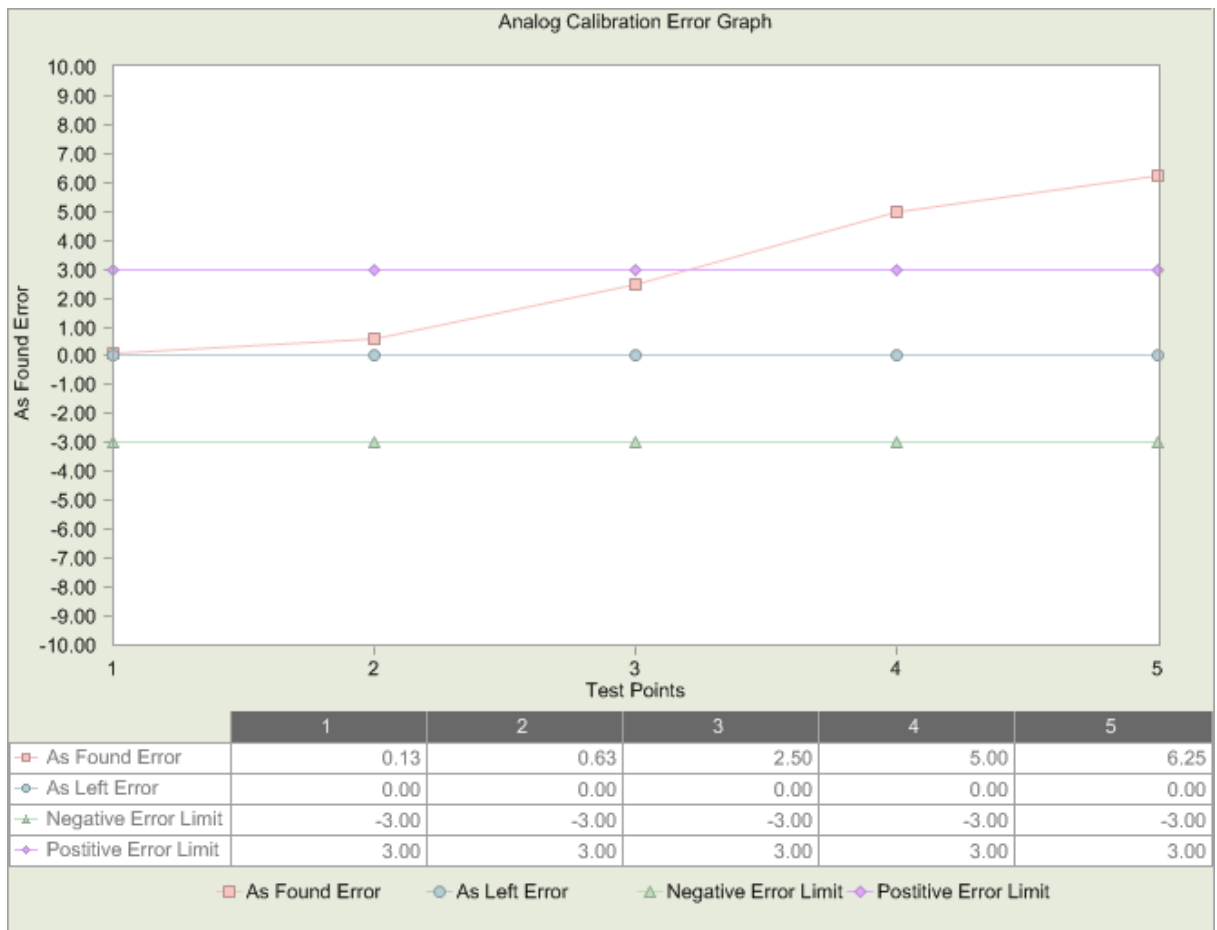
Changing the Graph Type

If desired, you can change the graph type that you are currently viewing and display the data in a different graph type. For example, if the data is being displayed in a bar graph, you can change the graph so that you are viewing the information in a pie graph.

To change the graph type:

1. In the Meridium APM Framework, on the **Graph Viewer** page, on the standard toolbar, click the **Gallery** button .

   A submenu appears, displaying all graph types.

2. Click the icon representing the desired graph type.

   The graph reloads, and the data is displayed in the selected style.

   > **Note:** If you select a graph type in the list, click the **Apply** button, and nothing happens, it means that the data in the underlying query or dataset does not meet the requirements of the selected graph type. You should select a different graph type or modify the data to fit the desired graph type.

3. On the **Common Tasks** menu, click the **Save Graph** link.

   Your changes are saved. The next time that you view the graph, you will see your changes.

Modifying the Title of a Graph

The graph title is displayed at the top of the graph in the graph workspace. Note that the title property is different from the name and caption of the graph, which you can manage via the **Graph Properties** dialog box or via the Meridium APM Catalog. By default, when you create a new graph, the graph title is set to the Catalog item caption of the query or dataset upon which the graph is based. But you can change the graph title if desired.

To modify the title of a graph:

1. In the graph workspace, right-click the current graph title, and then click **Edit title**.

   The graph title appears in an editable text box.

2. Modify the title as desired.

3. Click anywhere outside the text box.

   The new title appears in display mode.

4. On the **Common Tasks** menu, click the **Save Graph** link.

   Your modifications to the title are saved.

Modifying the Name, Caption, and Description of a Graph

Graphs are stored as items in the Meridium APM Catalog. As with all Catalog items, you can modify the graph properties via the Catalog itself. Additionally, via the **Graph Viewer** page, you can view the Catalog item properties and modify the name, caption, and description.

> **Note:** The name and description are associated with the Catalog item itself and are different from the graph title.

To change the name, caption, and description of a graph:

1. In the Meridium APM Framework, on the **Graph Viewer** page, on the **Graph Tasks** menu, click the **View Properties** link.

   The **Graph Properties** dialog box appears.

2. Modify the ID (i.e., Catalog item name), caption, and description as desired. If desired, click the 🌐 button to [manage translations for that string](#).

> **Note:** The remaining properties are read-only.

3. Click **OK**.

   The **Graph Properties** dialog box closes.

4. On the **Common Tasks** menu, click the **Save Graph** link.

   Your changes are saved and the new name of the graph appears in the title bar.

Making Prompts Appear at the Top of the Graph Viewer Workspace

Suppose that your database contains a graph that illustrates the amount of lost profit that each production event caused, and a prompt is defined on the underlying query to allow you to specify the associated production event code.

In this case, if you opened the graph from the Catalog, you would see the following prompt in a stand-alone dialog box, like the one shown in the following image.
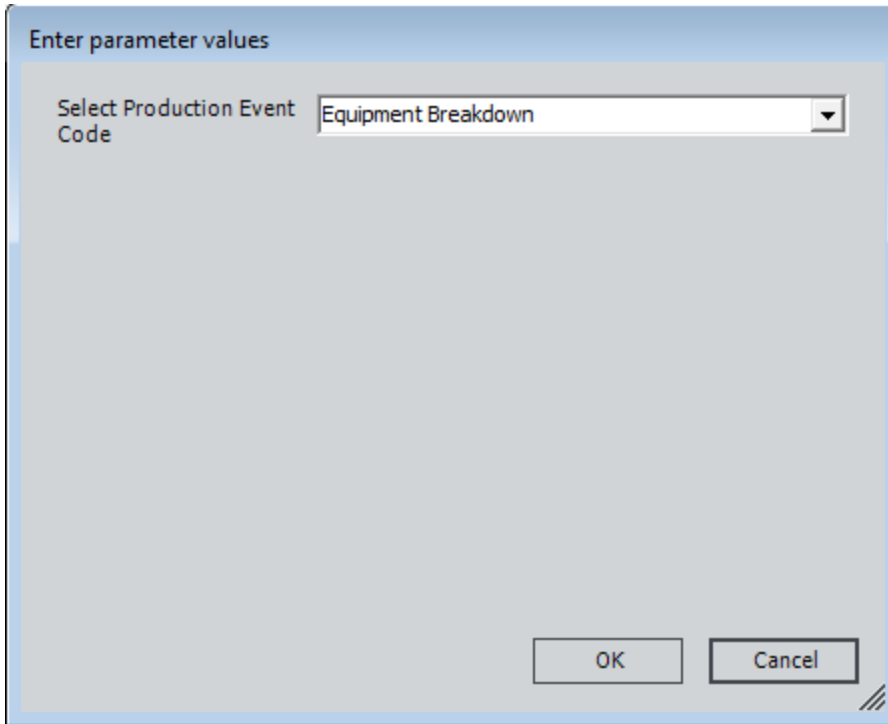
You would need to provide a value in the prompt and click **OK** in order to see the results on the graph, as shown in the following image.

If you wanted the prompt to appear at the top of the workspace, however, you could access the graph from a URL similar to the following one:

meridium://Registry/GraphViewer?GraphPath=Public\Lost Profit by Event - Event Code Drill Down Graph&PromptHeader=1

After you executed the URL, the graph would appear with the prompts at the top of the workspace, as shown in the following image.

You could select the desired value in the prompt and then click the **Run Query** button to display the updated graph below the prompt, as shown in the following image.

Making Prompts Appear at the Top of the Graph Viewer Workspace
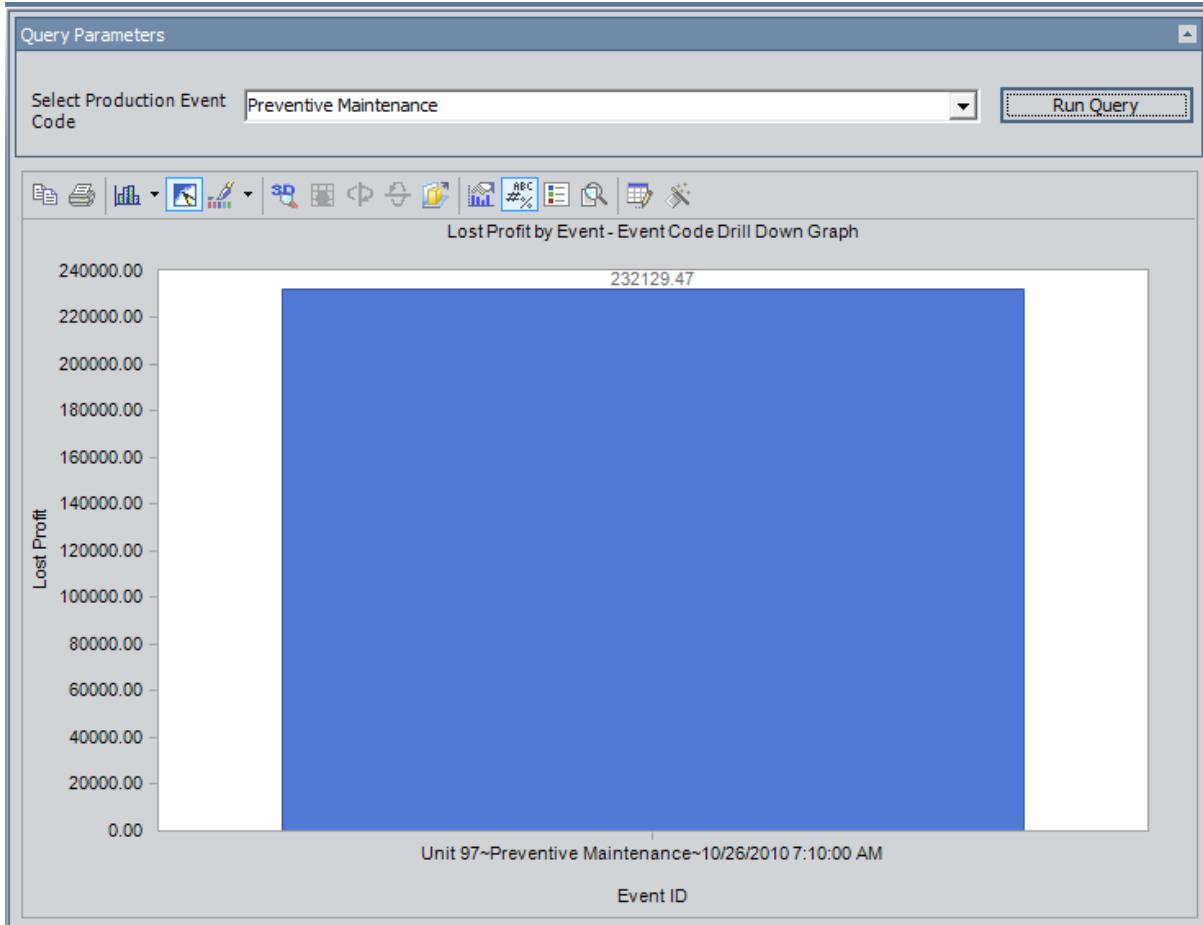
Using this approach, you can continue to change the values in the prompts and rerun the underlying query to refresh the graph. This approach is useful if you want to be able to see the graph results along with the values that were used to yield those results.

Confidential and Proprietary Information of Meridium, Inc. • Page 427 of 961

Refreshing Graphs

You can refresh the **Graph Viewer** page to ensure that the graph is displaying the most current database information and configuration options. For example, you might want to refresh the graph to rerun the associated query or to see the effect of adding a record to the database.

> **Note:** If the underlying query contains a prompt, the prompt will be displayed after you refresh the graph. You will need to supply the desired prompt values again.

To refresh the graph:

- In the Meridium APM Framework, on the **Graph Viewer** page, on the **Graph Tasks** menu, click the **Refresh Graph** link.

  The graph refreshes to display the most current information.

Viewing the Query or Dataset Associated with a Graph

To view or modify the query or dataset that was used to create a graph:

- In the Meridium APM Framework, on the **Graph Viewer** page, on the **Graph Tasks** menu, click the **Go To Query/Dataset** link.

  The query or dataset appears. At this point you can modify the query or dataset, and then save the changes. The next time that you view the graph that was created based on the query or dataset, you will see your changes.

Copying Graphs

If desired, you can copy a graph as an image or as text only to your clipboard and then paste the data to use in another location. You can also [export the graph](#).

To copy a graph:

1. In the Meridium APM Framework, on the **Graph Viewer** page, on the standard toolbar, click the **Copy** button ⧉ .

   A submenu appears.

   The submenu contains the following options:

   - **As a Bitmap:** Lets you copy the graph as a .BMP image.
   - **As a Metafile:** Lets you copy the graph as a Windows .WMF image.
   - **As Text (data only):** Lets you copy the graph as a .TXT file.

2. Click the desired option.

   The graph is copied to the clipboard as the selected file type. You can use the data in another location. For example, if you copy the graph as a .TXT file, you can paste the data into Notepad.

Exporting Graphs

When you export a graph, you extract its contents from the Meridium APM system and save it to an external file.

To export a graph:

1. In the Meridium APM Framework, on the **Graph Viewer** page, on the **Graph Tasks** menu, click the **Export Graph** link.

   The **Export Graph to File** dialog box appears.
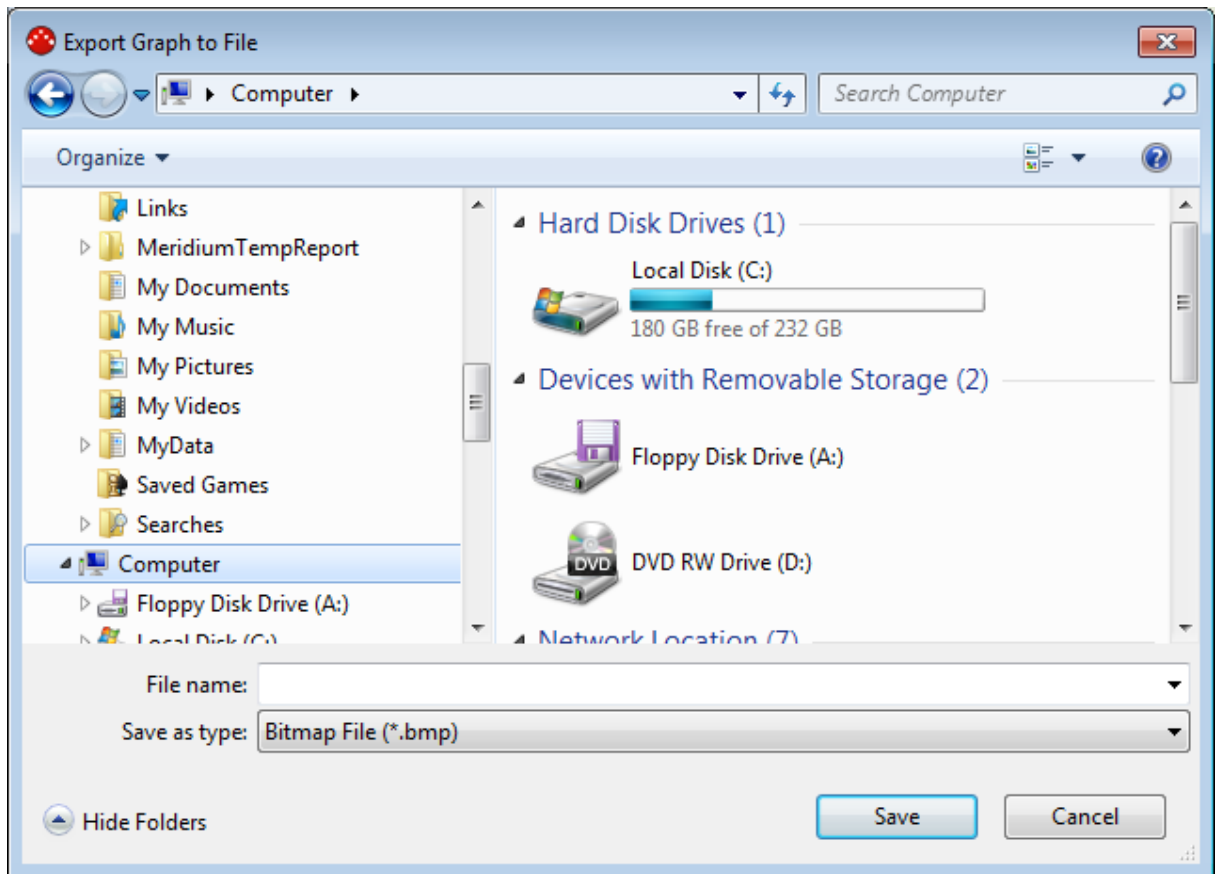
2. Navigate to the desired location.

3. In the **File name** text box, type a name for the file.

4. In the **Save as type** list, select the file type that you want to use. You can save the graph as:

   - **Bitmap File (*.BMP):** Saves the graph as a bitmap image.
   - **Windows Metafile (*.WMF):** Saves the graph as a Windows metafile image.

- **Text File (*.TXT):**Saves the data in the graph as a text file.

5. Click the **Save** button.

   The graph is saved to the selected location.

Saving Graphs

After you [create a graph](#), you can save it so that you or other users can access it again later. Graphs are saved as Catalog items to the Meridium APM Catalog. After a graph has been saved to the Meridium APM Catalog, whenever you make changes to it, you will save to re-save the graph to retain your changes.

To save a graph to the Meridium APM Catalog:

1. While viewing the graph on the **Graph Viewer** page, on the **Common Tasks** menu, click the **Save Graph** link.

   If the graph has already been saved, any changes that you have made will be saved to the current Catalog folder and with the existing Catalog item properties. If the graph has not yet been saved, the **Save Graph** dialog box will appear, as shown in the following image. The following instructions assume that you are saving the graph for the first time.
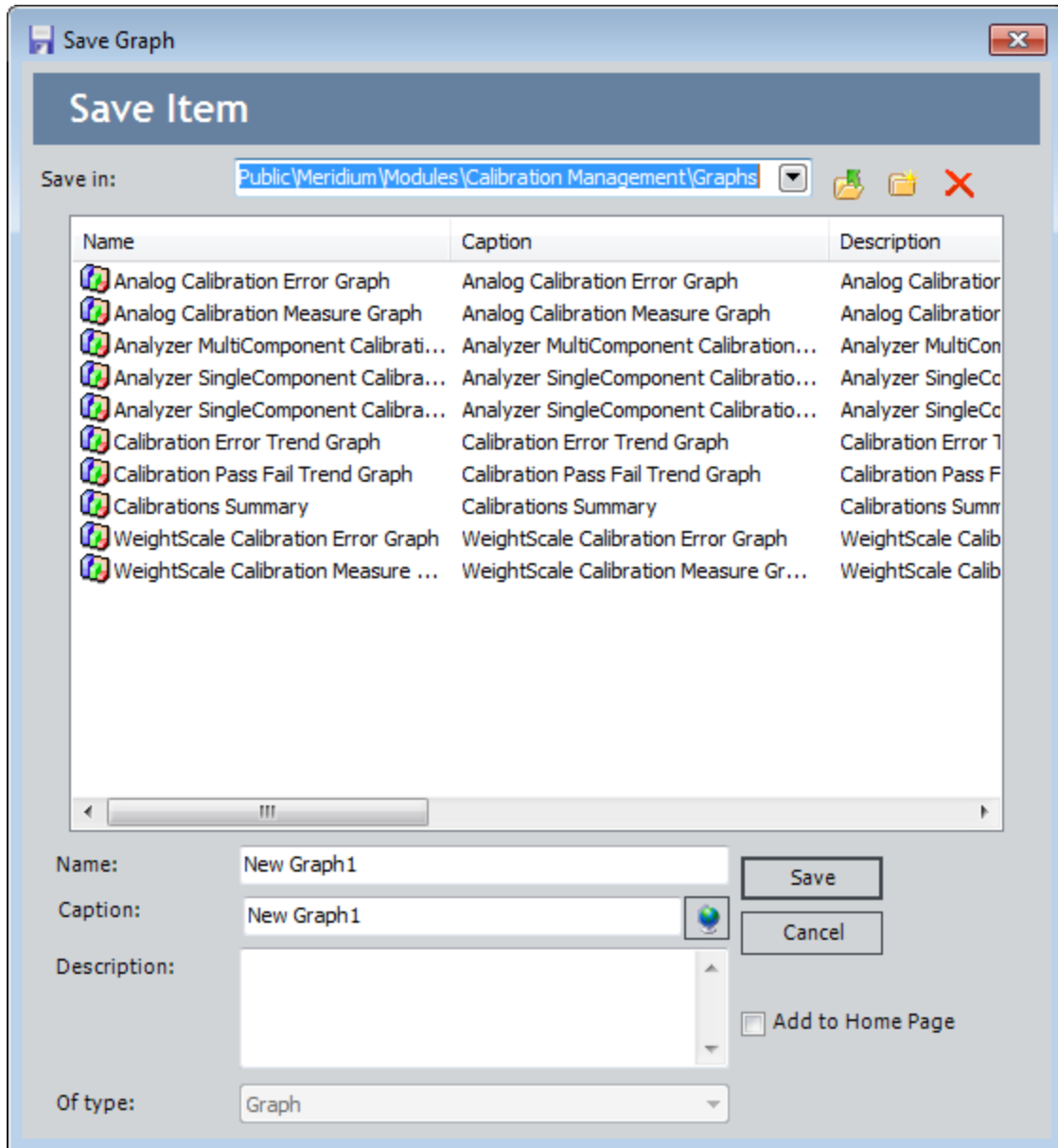
> **Note:** If you are viewing a saved graph on the **Graph Viewer** page and you click the **Save As** link on the **Common Tasks** menu, the **Save Graph** dialog box will appear so that you can save a *copy* of the current graph with a different name or to a different location. You can use the following instructions to complete that task, as the process is the same as saving a graph for the first time.

2. Navigate to the Catalog folder in which you want to save the graph.

3. In the **Name** text box, type a name for the graph. The name is required and must be unique to the Catalog folder in which you are saving the graph.

The **Caption** text box is populated automatically with the value that you typed in the **Name** text box.

4. If desired, click the [🌐] button to manage translations for that string.

5. If desired, type a description of the graph in the **Description** text box. This is not required to save the graph.

6. If desired, select the **Add to Home Page** check box if you want toadd a link to the graph to a Home Page. This is not required to save the graph.

7. Click the **Save** button.

   If you:

- Selected the **Add to Home Page** check box, the graph is saved to the Catalog, and the Send to Home Page - Select Home Page dialog box is displayed.

   Depending upon your membership in the MI Power User Role Security Group and whether or not you are a Super User, you can use the **Send to Home Page - Select Home Page** dialog box provide a link to the dashboard to your Personal Home Page, a Group Home Page, or the Personal Home Page for a particular Security User.

- Did not select the **Add to Home Page** check box, or you have completed the process of adding the graph to a Home Page, the **Save Graph** dialog box closes, and the graph is saved to the Catalog.

Deleting Graphs

To delete a graph from the Catalog:

1. In the Meridium APM Framework, on the **Graph Viewer** page, on the **Common Tasks** menu, click the **Delete Graph** link.

    > **Note:** This link is enabled only if you are viewing a graph that has been saved to the Catalog.

    A confirmation message appears, asking if you are sure that you want to delete the graph.

2. Click the **Yes** button.

    The graph is deleted from the Catalog.

Accessing the Query Builder Page

To access the Query Builder page:

- [Begin a new query](#).

  -or-

  Open an existing query from the Catalog.

  The **Query Builder** page appears.\

- If you began a new query using the **Query Builder**, the query design appears, reflecting the options that you selected in the **Query Builder**.
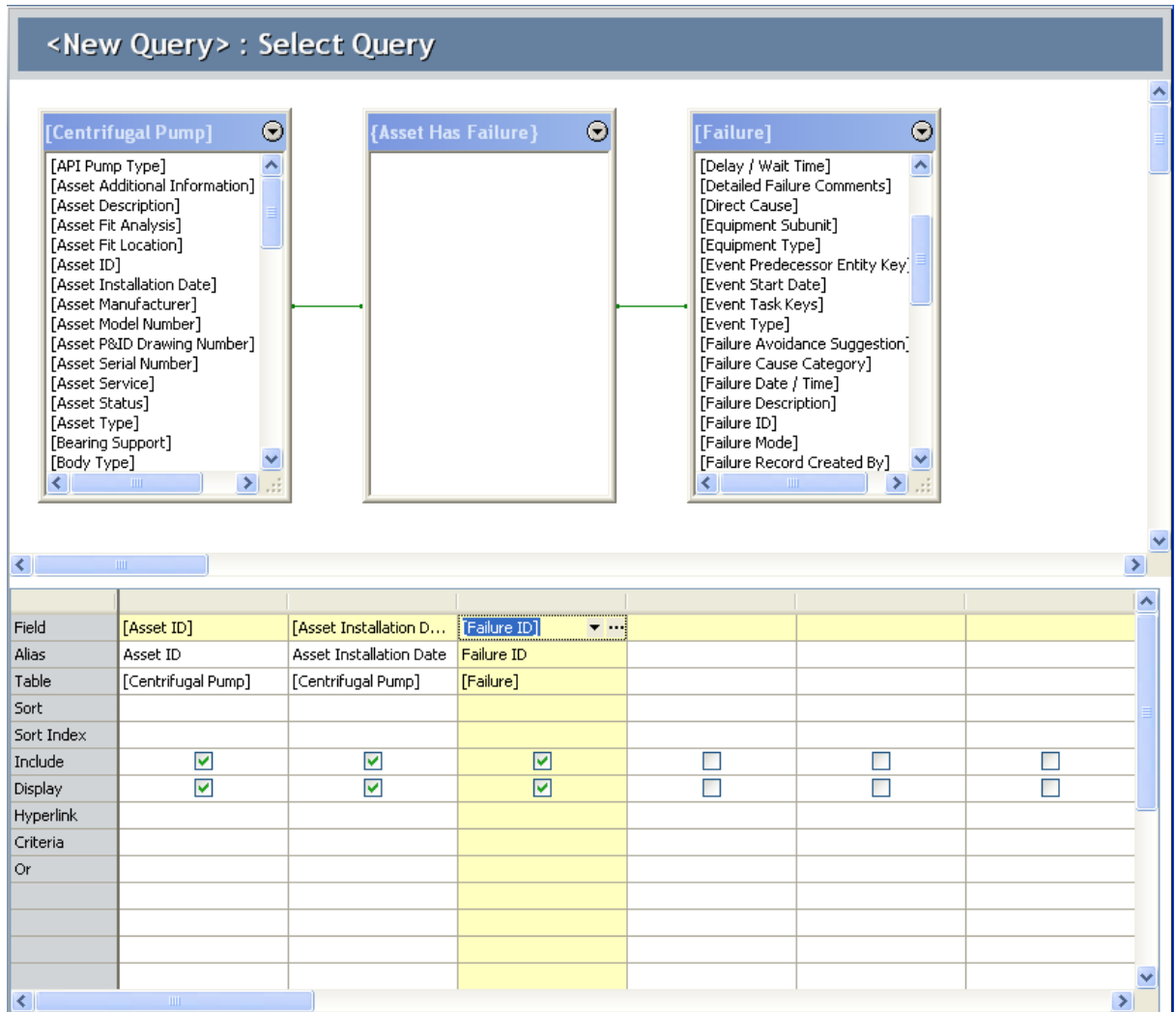
  -or-

- If you began a new query without using the **Query Builder**, the query design appears, reflecting the families you selected in the **Add Query Source** window.

-or-

■ If you opened an existing query from the Catalog, the query results appear.

Aspects of the Query Builder Page

The **Query Builder - <Query Caption>: <Query Type>** page appears when you open an existing query or [create a new query](#), where **<Query Caption>** is the caption of the query, and **<Query Type>** is the type of query.

If you are viewing a new query that you have not yet saved, the page title will be **Query Builder - <New Query>: Select Query** until you save the query. After you save the query, the title will be updated, where:

- **<New Query>** will be replaced with the caption of the query as it is stored in the Catalog item.

- Select Query will be replaced with the type of query if you have changed the query type (e.g., *Update Query*).

For example, in the following image, the caption of the Select query is *Equipment*.

Throughout this documentation, we refer to this page as simply the *Query Builder page*.

The **Query Builder** page displays different features depending upon whether you are viewing the query design, the SQL code, or the query results.

Regardless of whether you are viewing the query design, the SQL code, or the query results, the **Query Builder** page always contains the following task menus, where the links that are available on the menus change depending upon the context of the page.

- Query Tasks
- Common Tasks
- Other Tasks

The Query Design

The *query design* is a visual representation of the families, fields, and criteria that the query is configured to return. When designing a query, you can customize it to return specific records using specific formatting and criteria. As you design a query, SQL code is generated to store the code that is represented by the visual design.

The query design appears you click the **Design View** link on the **Query Tasks** menu while you are viewing query results or SQL code.

When you design a query, you can use the following areas to add sources, fields, and query criteria:

- Design canvas
- Design grid

The following image shows an example of a query design.

Design Canvas

The design canvas appears at the top of the query design and is the area where you can view and manage query sources. The following image shows an example of what the design canvas looks like when it includes two related families, Air Cooled Heat Exchanger and Failure, and the associated relationship family, Asset Has Failure.
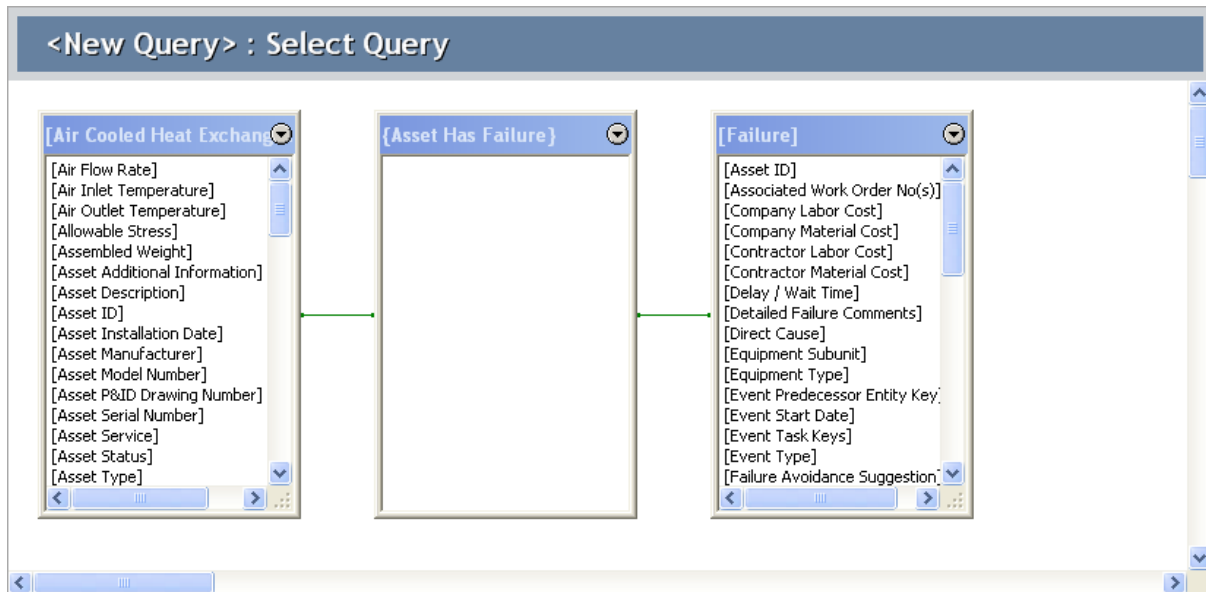
> **Note:** You can rearrange where the query source lists are displayed on the design canvas by dragging them to the location where you want them to appear.

Each query source list has a ⊙ button in the upper right corner, which you can click to display a menu with the following options:

- **Show Fields:** Displays a submenu with options that let you determine which fields to show in the query source list.
    - **Show All Fields:** Displays all the fields that are defined for the selected family. This option is selected by default.
    - **<Datasheet>:** Displays only the fields that exist on the datasheet identified by the name of the menu option. The number of **<Datasheet>** options corresponds to the number of datasheets that are defined for the family.
        - **Show System Fields:** Displays the system fields associated with the family.
        - **Show Inactive Fields:** Displays the inactive fields that exist in the family.
- **Link With:** Displays a submenu that lets you link the selected predecessor or successor family to a specified successor or predecessor family through the defined relationship. This option is enabled only when you add query sources in a specific

order, such that Meridium APM cannot determine how you want to relate the families due to multiple available relationship options. This option is hidden when you are working with a relationship family.

- **Add Related Family:** Displays the **Add Related Family to [<query source>]** window, which contains a grid that displays the list of entity and relationship families that are used in relationship definitions with the selected query source. Each row in the grid represents one relationship definition in which the selected query source participates as either a predecessor or a successor. By default, this list is sorted alphabetically by the entity family name. You can use this window to add to the query the families that are used in a relationship definition with any query source that is already displayed on the design canvas.

  The **[<query source>]** label in the title bar on this window will vary to match the name of the query source from which you accessed the **Add Related Family To [<query source>]** window.

  The **Add Related Family** link is hidden when you are working with a relationship family as a query source.

- **Remove:** Removes the selected query source from the design canvas.
- **Properties:** Displays the **Query Source Properties** dialog box, where you can define a query source alias.

Design Grid

The design grid appears at the bottom of the query design below the design canvas.

Field	▼ ···				
Alias					
Table					
Sort					
Sort Index					
Include	☐	☐	☐	☐	☐
Display	☐	☐	☐	☐	☐
Hyperlink					
Criteria					
Or					

The design grid is divided into rows and columns. Each column represents a field that will be included in the output. For example, in the following image, the columns that will appear in the output are Asset ID, Asset Description, and Asset Installation Date.

Field	[Asset ID]	[Asset Description]	[Asset Installation Date]
Alias	Asset ID	Asset Description	Asset Installation Date
Table	[Actuator]	[Actuator]	[Actuator]
Sort			
Sort Index			
Include	☑	☑	☑
Display	☑	☑	☑
Hyperlink			
Criteria			
Or			

You can adjust the width of the columns for viewing purposes by placing your mouse pointer on the column border and dragging the divider until the column reaches the desired width. Alternatively, you can double-click a column divider to expand the column to the width of the largest row. The column highlighted in yellow represents the currently selected column. As you scroll across the design grid, the first column containing the rows Field, Alias, Table, Sort, and so on remains stationary so that you can always see the row for which values are defined.

For each column, the design grid displays the following rows:

- **Field:** The name of the family field.

You can add or modify a field by clicking the down arrow and selecting the desired field. Alternatively, you can drag a field from any query source into a field cell of the design grid. You can remove a field by right-clicking it and choosing **Delete**.

- **Alias:** The alternate name for a given field. The alias will appear as the column heading in the query output.

- **Table:** The name of the family to which a field belongs.

- **Total:** A value that specifies how output will be grouped: **Group By**, **Sum**, **Avg** (average), **Min** (minimum), **Max** (maximum), **Count**, **Expression**, or **Where**. You can also create aggregate functions for fields using the Expression Builder.

> **Note:** This row appears by default for Crosstab queries. To make this row appear for Select queries, on the **Query Tasks** menu, click **Options**, and then click **Show Totals**.

- **Crosstab** (appears for Crosstab queries only): The cross-tabulation setting for the field, whether the field will be a **Row Heading**, **Column Heading**, or **Value**.

- **Sort:** The sorting method that will be used for this column in the output: **Ascending**, **Descending**, or **(not sorted)**.

- **Sort Index:** A value that indicates the order in which sort criteria will be applied. For example, the first field to which you apply sort criteria receives a Sort Index of 1; this is the value on which results will be sorted first. The second field is assigned a Sort Index of 2, the third a 3, and so on.

- **Include** (appears for Select queries only): A flag that specifies whether or not the field will be included in the SQL code. If this check box is selected, the field will be included in the SQL code.

- **Display** (appears for Select queries only): A flag that specifies whether or not the field will appear in the results. If this check box is selected, the field will appear in the results.

- **Hyperlink** (appears for Select queries only): The URL that will be used to generate links for this field. If a ,URL has been defined, when you run a query, the content of the column will appear as a hyperlink.

- **Criteria:** The conditions that will be applied to the query results. Note that criteria for a field must be appropriate for the data type of that field.

- **Or:** Another set of conditions that will be applied to the query results.

SQL Code

If you are familiar with SQL syntax, instead of designing the query, you can type *SQL code* directly and run the query to view the results. Meridium APM supports the use of Oracle schemas and SQL Server databases, both of which can be queried using SQL statements. While the same basic SQL code can be used to query either type of database, there are some differences in the syntax that is supported by each database server. Therefore, Meridium APM, Inc. has developed its own proprietary version of SQL that is constructed automatically and translated by the system at runtime into the SQL syntax that is appropriate for the type of database you are using. We call this form of SQL *Meta-SQL*.

In most cases, Meta-SQL syntax is the same as standard SQL syntax. This means that in most cases, you can type the SQL syntax you are familiar with. If, however, you use functions that are specific to one database server (e.g., Oracle), when you run the query on a different database server (e.g., SQL Server), an error appears, and you will be unable to run the query and view its results until you correct the SQL code.

In other words, when you type SQL code directly and click the **Run Query** link, the following things occur:

- The Meridium APM system reads the syntax and validates whether or not it is valid Meta-SQL.
- The Meridium APM system translates the Meta-SQL into SQL that can be interpreted by SQL Server or Oracle, depending on the type of database you are using.
- SQL Server or Oracle executes the translated code and returns results to the Meridium APM system.
- The query results appear.

Throughout the Meridium APM Framework documentation, we use the term *SQL* when referring to the SQL code that appears when you click the **SQL View** link on the **Query Tasks** menu.

The following image shows an example of SQL code.
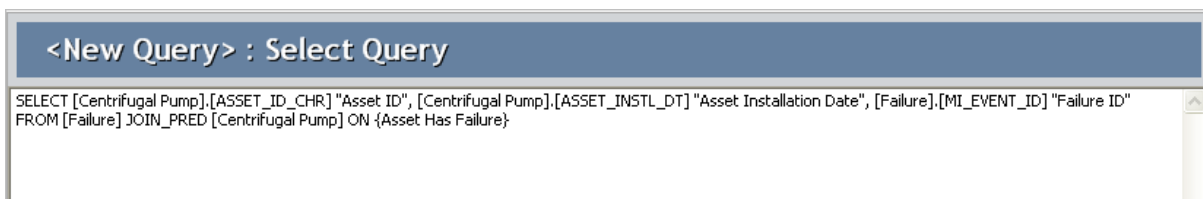
All tasks that you can perform when designing a quuery write Meta-SQL code that can be viewed when you click the **SQL View** link. However, not all SQL code that you can type directly can be interpreted by the query design. This means that in some cases, you can write SQL code that will cause the design to be unavailable. If you write a query using SQL code and want other users to be able to modify the query design, be sure to access the design to make sure it is available before saving the query.

Specific instructions for writing SQL code are beyond the scope of this documentation. Where appropriate, we provide guidelines and suggestions for how to write SQL expressions and use SQL functions, but this documentation does not contain comprehensive SQL code explanations.

Query Results

Query results appear when you:

- Open a Select or Crosstab query.

  -or-

- Click the **Run Query** link on the **Query Tasks** menu while viewing the query design or SQL code.

If you open a saved query using a hyperlink and the query contains prompts, if the URL behind the link contains the appropriate [parameter](#), you will see at the top of the results the possible parameter values, along with a **Run Query** button and a **Stop** button. You can select one or more different parameters values by which to limit the results and click the **Run Query** button to view the results of your selection.

The following links appear at the bottom of the results to help you navigate your search results:

- **Display record count:** Displays the number of total records returned by the query in the format **<#> records found**. So, if the query returns 44 records, and you click the **Display record count** link, it will display the text **44 records found**.

  To re-display the **Display record count** link, you can run the query again.

  Note that when you run the query, the text **[#] records found** appears by default if the number of records found is less than or equal to the number in the **Page Size** box.

- **First:** Displays the first page in the results set.
- **Previous:** Displays the previous page in the results set.
- **Next:** Displays the next page in the results set.
- **Last:** Displays the last page in the results set.
- **Page Size:** Lets you specify how many results you want to display on a single page. The default value is 1000, but you can specify a different number if desired.

  > **Note:** The greater the number of query results, the longer it will take for the results to appear on a single page.

In addition, when viewing the results, you can customize the layout of the results grid. Changes that you make to the grid layout while viewing the results are independent from customizations stored within the query itself. When you save a query when viewing the results, any changes that you have made to the grid layout will be saved as a personal layout for the current user.

You can also copy data from a cell in the results and paste it into a Meridium APM dataset, a cell on any query design grid, or an external location such as a Microsoft Excel spreadsheet. To copy the data from a cell in the results, highlight the data, and press Ctrl+C on your keyboard, or highlight the data, right-click the cell, and click **Copy**. You

can paste the data by pressing Ctrl+V on your keyboard or by right-clicking the desired location and clicking **Paste**.

Query Tasks Menu
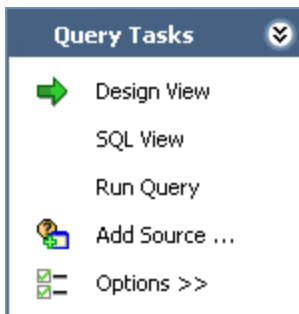
The **Query Tasks** menu that appears on the **Query Builder** page when you are viewing the query design contains the following links:

- **Design View:** Displays the query design, which provides a visual representation of how query source tables are set up and allows you to specify criteria for the results. When you are viewing the query design, a green arrow appears next to the **Design View** link.

- **SQL View:** Displays the meta-SQL code of the query. If you change the SQL code, the query design will reflect the changes.

- **Run Query:** Runs the query and displays the results. When you click this link, it becomes the **Stop Query** link, which, when clicked, will discontinue the query processing.

- **Add Source:** Displays the **Add Query Source** dialog box, where you can add another entity family, relationship family, or existing query as a source for the current query. This link appears only if you are creating a Select, Crosstab, or Append query.

- **Target Source:** Displays the **Target Query Source** dialog box, where you can specify which query source contains the records that you want to modify or delete. This link appears only if you are creating an Update, Append, or Delete query.

- **Options:** Provides a submenu that contains the following options:

  - **Unique Records Only:** Limits the results to records that contain unique values in the fields included in the query. This link appears only for Select, Update, Append, and Delete queries.

  - **Limit Results Set:** Displays the **Limit Results Set** dialog box, where you can specify the number of results that should be returned. This option is available only when the query is a Select query.

  - **Show Totals:** Adds to the design grid a **Total** row, which lets you specify that the output will be grouped by one of the following: **Group By**, **Sum**, **Avg** (average), **Min** (minimum), **Max** (maximum), **Count**, **Expression**, or **Where**. This link is enabled only for Select, Update, Append, or Delete queries.

- **Delete Force:** For Delete queries, specifies that the query will delete any records that meet the query criteria and are linked to other records. This link appears only for Delete queries.

- **Match Case:** Specifies whether or not any criteria you define for a field will be case sensitive. For example, if you define criteria to return only those records that contain the value *Active* in a given field, selecting this option will cause the results to include only records that contain the value *Active* using the same case that you typed in the **Criteria** cell. not selecting this option, however, will cause the results to include records with the value *Active* using any capitalization.

If you define criteria using the IN operator instead of the **=** operator, the criteria will *always* be case sensitive even if you do not select the **Match Case** option.

> **Note:** The **Match Case** option appears on this menu only when you are connected to an Oracle data source. By default, new queries constructed against an Oracle schema will be case sensitive, but you can modify the default case sensitivity if desired. This option is not available when you are connected to a SQL Server data source. Queries constructed against a SQL Server data source are *never* case sensitive.

- **Mode:** Displays a submenu with two options, **Formatted** and **Unformatted**, which lets you specify the query mode for Select queries. This link appears only for Select queries.

- **Sort Results Grid by Actual Value:** Specifies whether the query results will be sorted by the displayed value or the stored value.
  - When this option is selected, numeric columns will be sorted in the results grid by their *stored* value. A check mark appears to the left of this option to indicate that it is selected.
  - When this option is not selected, numeric columns will be sorted in the results grid by their *displayed* value. A check mark will not appear to the left of this option when it is not selected.

This option is available only for Select queries and applies only to numeric columns when the query mode is set to *formatted*. Queries running in unformatted mode will always be sorted by the *stored* values. This menu option is disabled when the query mode is set to *unformatted*.

> **Note:** This setting applies *only* to sorting that is applied on-the-fly in the query results. It does not apply to sorting that is applied via sort criteria, which always sorts results according to the stored value.

- **Change Page Size:** Displays the **Change Page Size** dialog box, where you can specify the number of results to return in the results. This option is available for all query types.

- **Select Query:** Specifies that the query is a Select query.

- **Crosstab Query:** Specifies that the query is a Crosstab query.

- **Update Query:** Specifies that the query is an <u>Update query</u>.
- **Append Query:** Specifies that the query is an <u>Append query</u>.
- **Delete Query:** Specifies that the query is a <u>Delete query</u>.

Common Tasks Menu
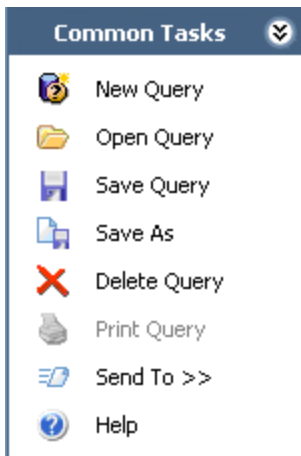
The **Common Tasks** menu that appears on the **Query Builder** page when you are viewing the query design contains the following links:

- **New Query:** Displays the **New Query** dialog box, where you can create a new query.
- **Open Query:** Displays the **Open Query** dialog box, where you can navigate to and open an existing query.
- **Save Query:** Saves the query you are current viewing. If a query has a saved personal or Super User grid layout, saving the query via the query design will cause those layouts to be discarded.
- **Save As:** Displays the **Save Query As** dialog box, where you can save a copy of the query with a different name and to a different location. If a query has a saved personal or Super User grid layout, saving the query via the query design will cause those layouts to be discarded.
- **Delete Query:** After asking for confirmation, deletes the current query. This link is enabled only when you are viewing a saved query.
- **Print Query:** This link is always disabled.
- **Send To:** Displays a submenu with options that let you provide a link to the current page on your desktop (create shortcut), in an email message, or on a Home Page. Note that this link is enabled only when you are viewing a saved query. Note also that if you viewing the results of a query that contains prompts, and you provide a link to the query to the Home Page, your desktop, or another user via email, the hyperlink that is created includes the values that you specified for the prompts.
- **Help:** Displays the context-sensitive Help topic for the current screen.

Other Tasks Menu

The links on the **Other Tasks** menu on the **Query Builder** page are *disabled* when you are viewing the query design. These links are enabled when you are viewing the [query results](#).

Query Tasks Menu
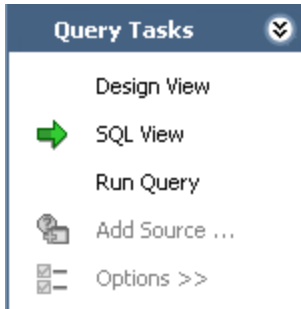
The **Query Tasks** menu that appears on the **Query Builder** page when you are viewing the SQL code contains the following links:

- **Design View:** Displays the query design, which provides a visual representation of how query source tables are set up and allows you to specify criteria for the results.

- **SQL View:** Displays the meta-SQL code of the query. If you change the SQL code, the query design will reflect the changes. When you are viewing the SQL code, a green arrow appears next to the **SQL View** link.

- **Run Query:** Runs the query and displays the results. When you click this link, it becomes the **Stop Query** link, which, when clicked, will discontinue the query processing. When you are viewing the query results, a green arrow appears next to the **Run Query** link.

- **Add Source:** This link is disabled when you are viewing the SQL code.

- **Options:** This link is disabled when you are viewing the SQL code.

Common Tasks Menu
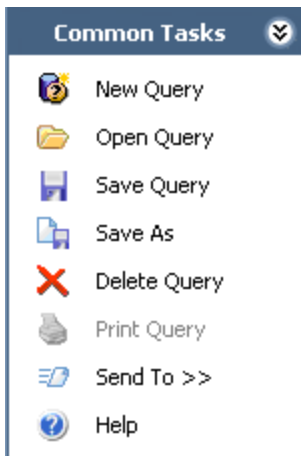
The **Common Tasks** menu that appears on the **Query Builder** page when you are viewing the SQL code contains the following links:

- **New Query:** Displays the **New Query** dialog box, where you can create a new query.
- **Open Query:** Displays the **Open Query** dialog box, where you can navigate to and open an existing query.
- **Save Query:** If a query has a saved personal or Super User grid layout, saving the query when you are viewing the SQL code will cause those layouts to be discarded.
- **Save As:** Displays the **Save Query As** dialog box, where you can save a copy of the query with a different name and to a different location. If a query has a saved personal or Super User grid layout, saving the query via the SQL code will cause those layouts to be discarded.
- **Delete Query:** This link is always disabled.
- **Print Query:** This link is always disabled.
- **Send To:** This link is always disabled.
- **Help:** Displays the context-sensitive Help topic for the current screen.

Other Tasks Menu

All options on the **Other Tasks** menu on the **Query Builder** page are *disabled* when you are viewing the query SQL code.

Query Tasks Menu
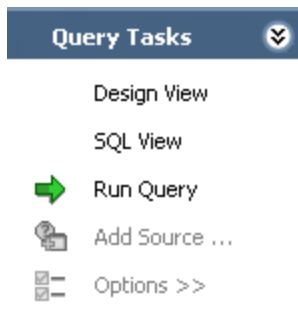
The **Query Tasks** menu that appears on the **Query Builder** page when you are viewing the query results contains the following links:

- **Design View:** Displays the query design, which provides a visual representation of how query source tables are set up and allows you to specify criteria for the results. This link does not appear when you access a query using a link whose underlying URL specifies view-only mode.

- **SQL View:** Displays the meta-SQL code of the query. If you change the SQL code, the query design will reflect the changes. This link does not appear when you access a query using a link whose underlying URL specifies view-only mode.

- **Run Query:** Runs the query and displays the results. When you click this link, it becomes the **Stop Query** link, which, when clicked, will discontinue the query processing. If the query contains prompts and was accessed via a hyperlink, this link does not appear if the URL specifies that the prompts should appear in the results. When you are viewing the query results, if the **Run Query** link is present, a green arrow appears next to it.

  Note that when you click the **Run Query** link, the page size is set to the default page size of 1000.

- **Add Source:** This link does not appear if you access a query using a link whose underlying URL specifies view-only mode. If you access a query using a link whose underlying URL does not specify view-only mode, this link is disabled when you are viewing the query results.

- **Options:** This link does not appear if you access a query using a link whose underlying URL specifies view-only mode. If you access a query using a link whose underlying URL does not specify view-only mode, this link is disabled when you are viewing the query results.

Common Tasks Menu
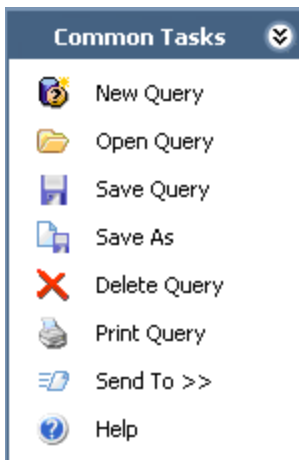
The **Common Tasks** menu that appears on the **Query Builder** page when you are viewing the query results contains the following links:

- **New Query:** Displays the **New Query** dialog box, where you can create a new query.

- **Open Query:** Displays the **Open Query** dialog box, where you can navigate to and open an existing query.

- **Save Query:** Saves the query you are current viewing. Additionally, the current grid layout for the query will be saved as a personal layout for the current user.

- **Save As:** Displays the **Save Query As** dialog box, where you can save a copy of the query with a different name and to a different location. Additionally, the current grid layout for the query will be saved as a personal layout for the current user.

- **Delete Query:** After asking for confirmation, deletes the current query. This link is enabled only when you are viewing a saved query.

- **Print Query:** Displays the **Preview** window, where you can print the results of the current query.

- **Send To:** Displays a submenu with options that let you . Note that this link is enabled only when you are viewing a saved query. Note also that if you viewing the results of a query that contains prompts, and you provide a link to the query to the Home Page, your desktop, or another user via email, the hyperlink that is created includes the values that you specified for the prompts.

- **Help:** Displays the context-sensitive Help topic for the current screen.
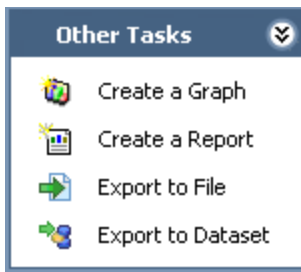
Other Tasks Menu

The **Other Tasks** menu that appears on the **Query Builder** page when you are viewing the query results contains the following links:

- **Create a Graph:** Launches the **Graph Setup** window, where you can create a graph based on the query you are currently viewing. Note that if the query does not contain any numeric fields, the **Graph Setup** window will appear, but you will not be able to define a series for the graph. You will, therefore, not be able to create a graph based on a query that does not contain any numeric fields.

- **Create a Report:** Lets you create a report based on the query you are currently viewing. This link is enabled only if the Microsoft SQL Server Business Intelligence Development Studio is installed on the Meridium APM Framework workstation.

- **Export to File:** Displays the **Query Results Export Wizard**, where you can export the query results to a text, Microsoft Excel, or XML file.

- **Export to Dataset:** Displays the **Save dataset** dialog box, where you can save the query results as a Meridium APM dataset.

About Creating Queries

You can create queries that are as simple or as complex as you need them to be, depending upon the data that you want them to return. For instance, a very simple query might return records from only one family and show the value in only one field in those records. A more complicated query, on the other hand, might return records from multiple families, convert values in several fields, and perform calculations on the converted values.

The process of creating a *simple* Select query includes the following steps:

1. Initiate the creation process.

2. Add at least one query source.

3. Add at least one field from that query source.

4. Run the query to make sure it returns the expected results.

5. Save the query.

The process of creating a more complicated Select query might include the following *additional* steps:

6. Add criteria using expressions.

7. Add prompts.

8. Add hyperlinks.

9. Define specific requirements of the results, such as how they will be sorted and the default page size.

When you initiate the query creation process, you can choose to select the query sources (entity and relationship families) and query columns (family fields) using either of the following options:

- **Query Builder:** Takes you step-by-step through the process of building a Select query. The **Query Builder** can be helpful to novice users who are not familiar with query syntax and logic. Also, if you want to create a query using two related families, the **Query Builder** will display the related families that exist for the selected family. Using the **Query Builder**, you can build a query based either on a family or on an existing query.
- **Query design:** Provides a visual representation of a query and lets you manually add sources, criteria, and links. The designer is intended for more advanced users.

After you have initiated the process of creating a Select query using either of these options, when the query design appears, you can modify the query type to create any of the following types of queries based upon that Select query:

- Crosstab query
- Delete query

- [Update query](#)
- [Append query](#)

Building a Query Based on a Family

A family-based query is a query that questions the database about a family or group of similar records. For example, you could create a family-based query to get information about all the equipment in the database. Note that you can also create a query based on an existing query.

To build a query based on a family:

1. In the Meridium APM Framework, on the Meridium APM Framework toolbar, click the **Query** button, and then click **Build New**.
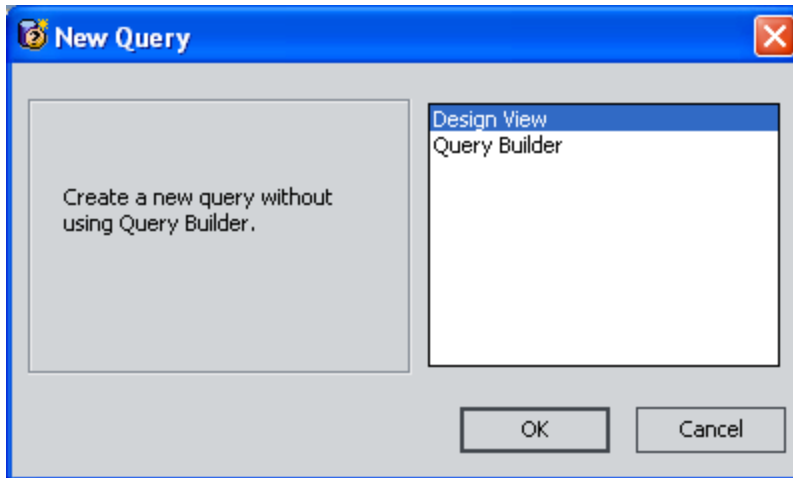
    The **New Query** dialog box appears.

2. Select **Query Builder**, and click **OK**.

    The **Query Builder** appears, displaying the first step in the creation process. The **Query is based on a family** option is selected by default.
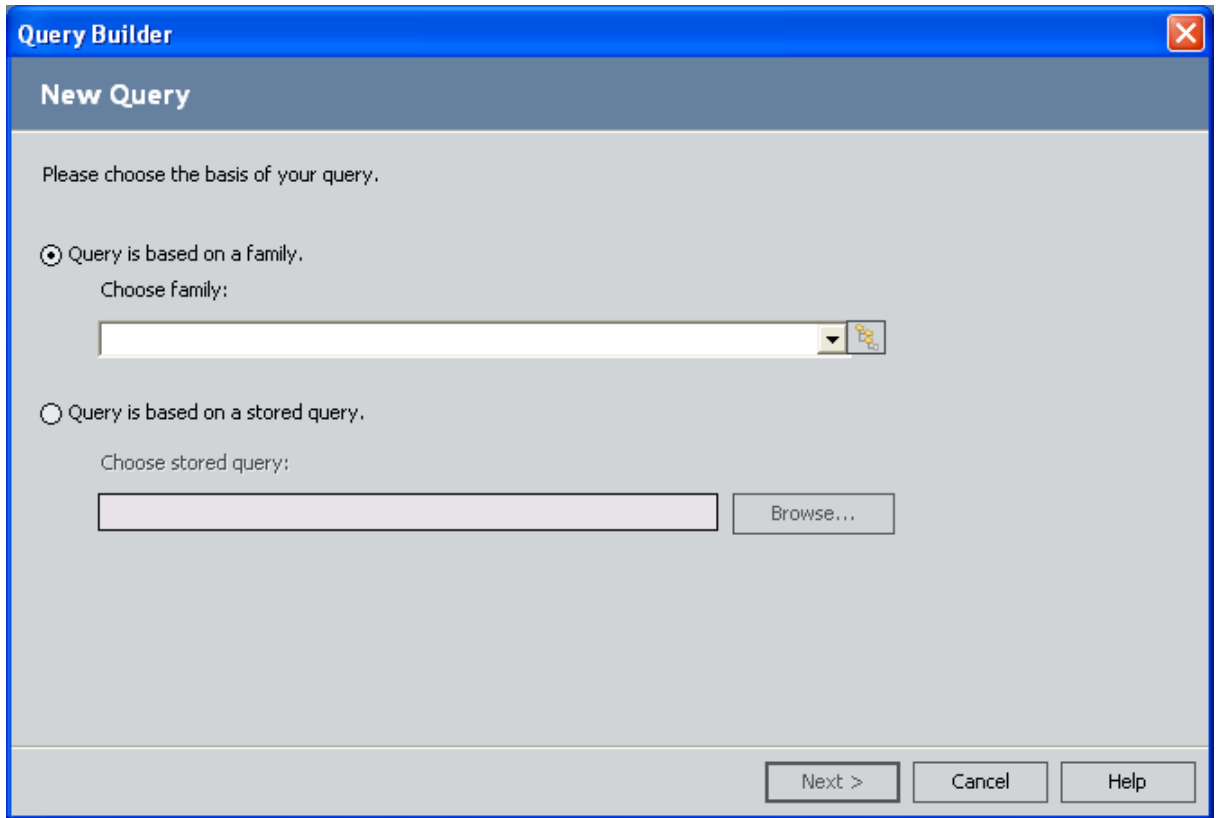
3. In the **Choose family** list, select the family on which to base your query. The list will display only families to which you have been granted at least View privileges in the Configuration Manager.
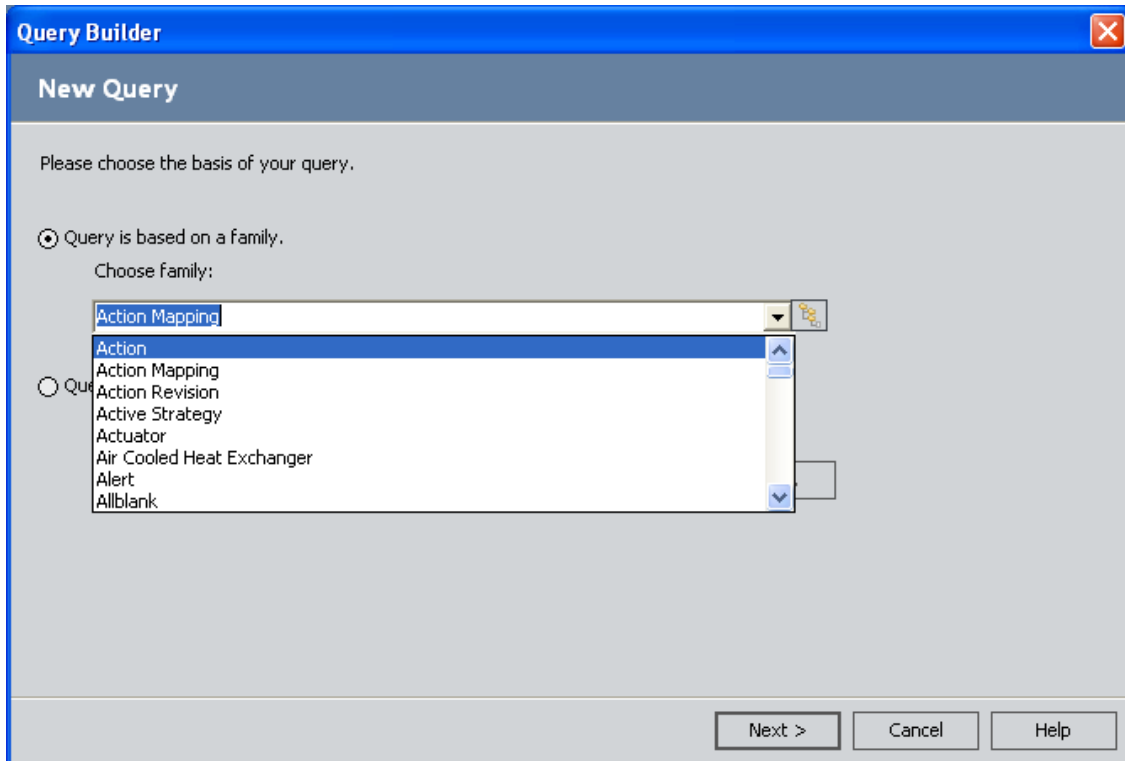
> **ŒHint:** Alternatively, you can click the [icon] button to make your selection from the hierarchical list of families. The list will display only families to which you have been granted at least View privileges in the Configuration Manager.

4. Click the **Next** button.

   The **Query Builder** displays the next step, which shows a list of families that are related to the family that you selected in the previous step.
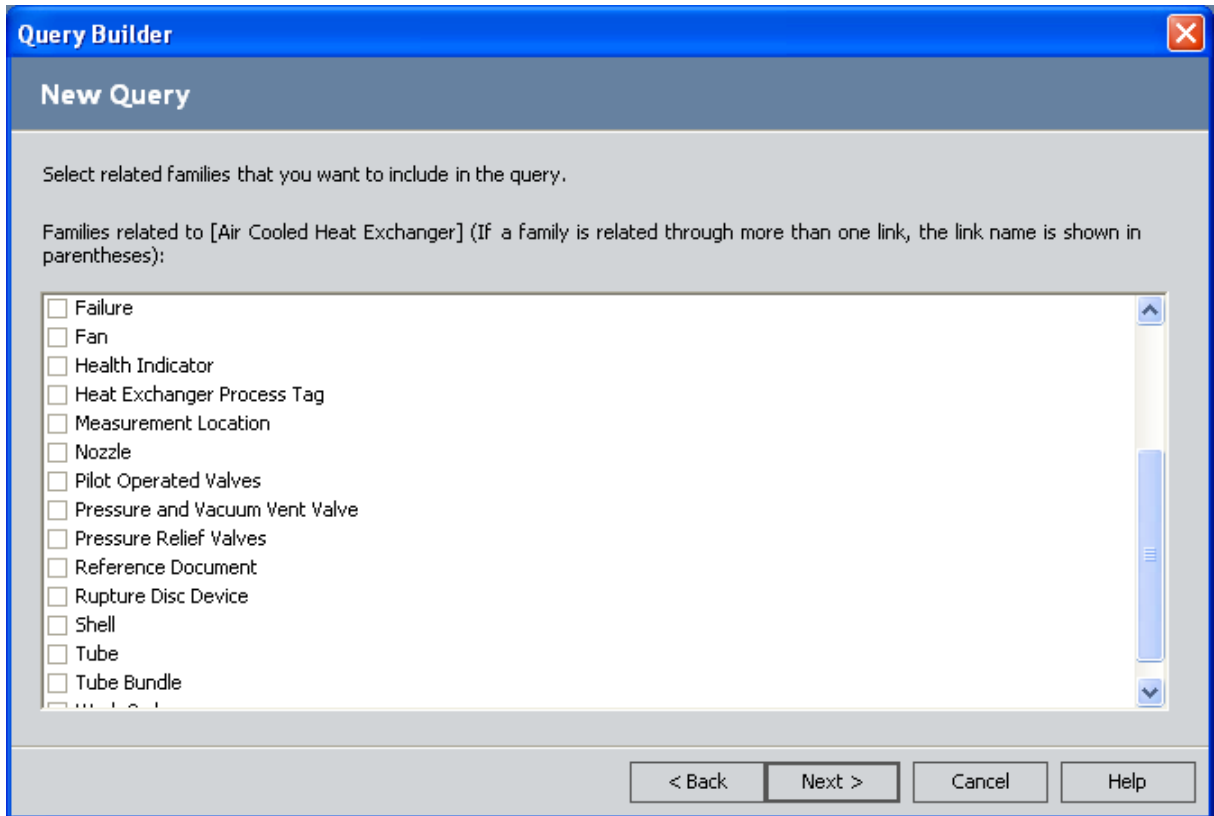
This screen displays a family only if you have at least View privileges to the relationship family that relates the family to the family selected on the previous screen. For example, suppose the Equipment family is related to the Task family through the Has Task relationship. If you selected the Equipment family on the previous screen, this screen would display the Task family only if you have at least View privileges to the Has Task relationship family.

5. In the list, select the check box next to each related family that you want to include in the query, and click the **Next** button. The list will display only families to which you have been granted at least View privileges in the Configuration Manager.

   The **Query Builder** displays the next step in the process, which shows a list of fields that exist for the selected families.
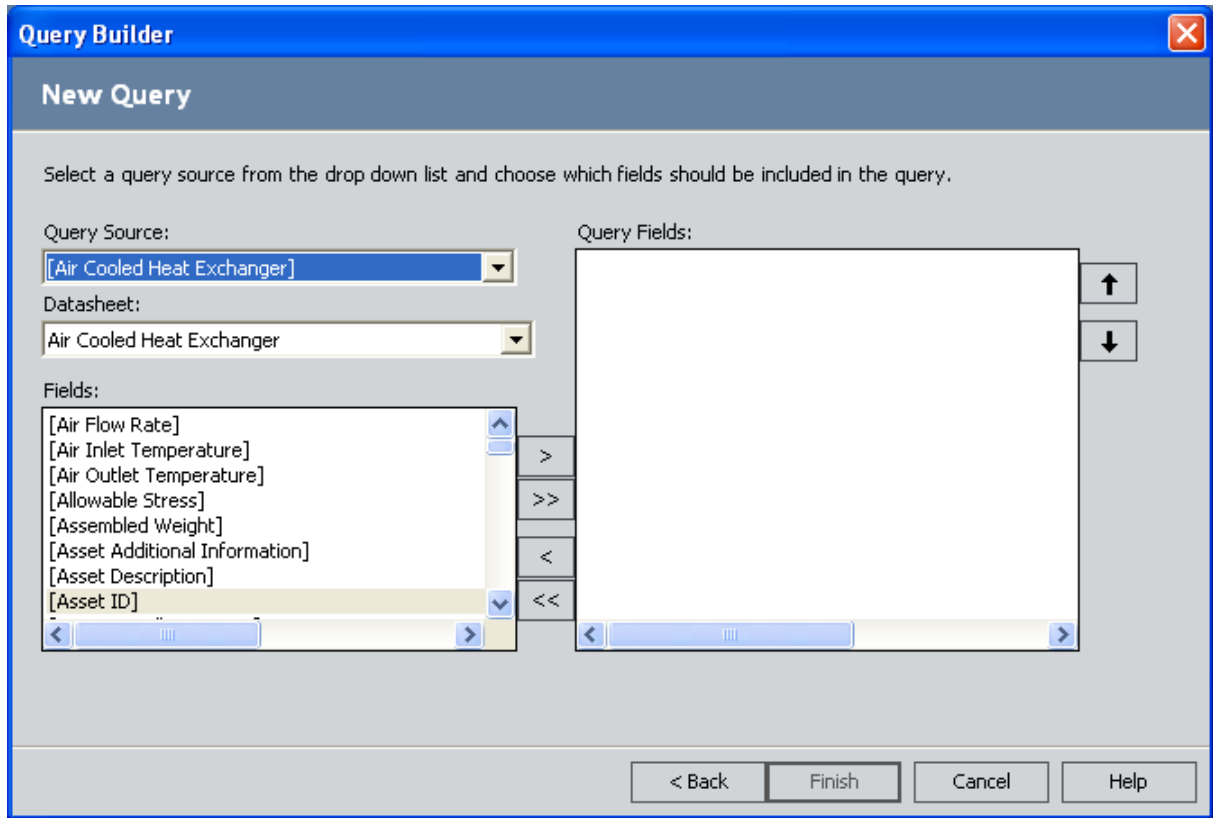
6. Select the fields that you want to include in the query. To do so:

   a. In the **Query Source** list, select a family.

   b. In the **Datasheet** list, select a datasheet. The available datasheets correspond to the family that you selected in the **Query Source** list. If you choose **<all fields>**, the **Fields** list will include all fields defined for the family. If you choose a specific datasheet, the **Fields** list will contain only those fields that exist in that datasheet.

   c. In the **Fields** list, select the desired fields, and click the right arrow button to add them to the **Query Fields** list. You can select multiple fields or click the double right arrow button to add all fields to the query. Similarly, you can use the left arrow and double left arrow buttons to remove fields from the query.

   > **Note:** System fields and inactive fields do not appear in the list of available fields. If you want to included system fields or inactive files in your query, you must use the query design.

   d. Repeat steps a through c for each query source.

   > **Note:** If you click the **Back** button at this point, any field selections that you have made will be lost. If you need to return to a previous screen in the **Query**

Builder to add additional query sources, you will need to re-add the desired fields to the query.

7. Click the **Finish** button.

   The query design for the new query appears.

8. To run the query, on the **Query Tasks** menu, click the **Run Query** link.

   -or-

   To save the query, on the **Common Tasks** menu, click the **Save Query** link.

Building a Query Based on an Existing Query

You can use an existing Select query as the basis for a new query to generate more specific results.

> **Note:** When creating a query that is based on another query, square brackets around the query name are required in the SQL code. The query name is case sensitive. When you use the **Query Builder** to construct a query based on another query, this syntax is provided automatically.

To build a new query based on an existing query:

1. In the Meridium APM Framework, on the Meridium APM Framework toolbar, click **Query**, and then click **Build New**.
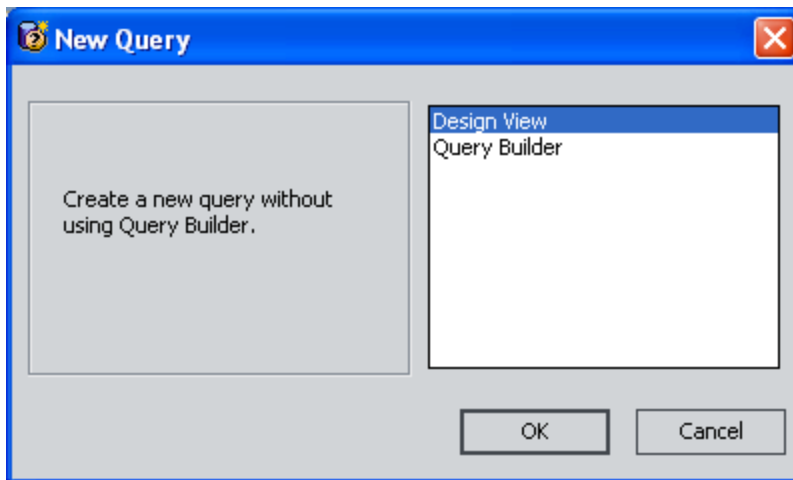
   The **New Query** dialog box appears.

2. Select **Query Builder**, and click **OK**.

   The **Query Builder** appears, displaying the first step in the creation process.
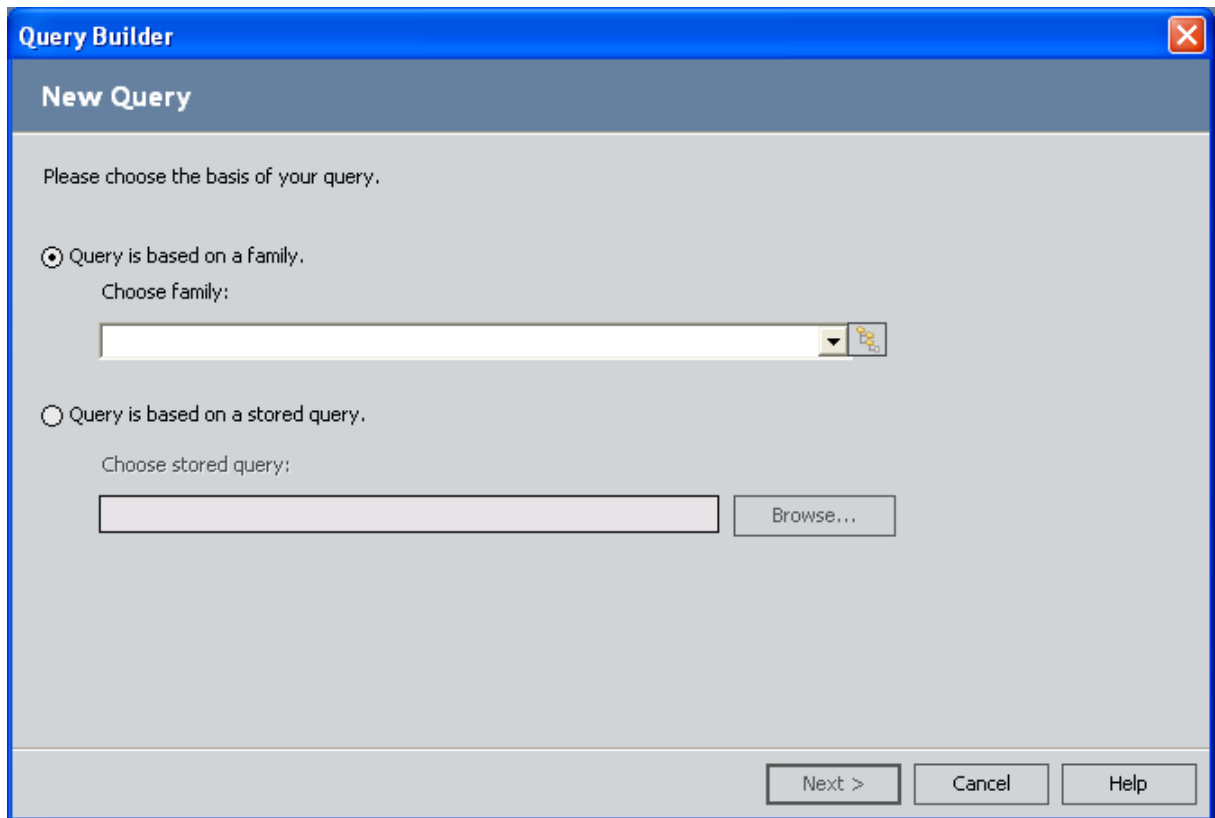
3. Select the **Query is based on a stored query** option.

    The **Choose stored query** field becomes enabled.
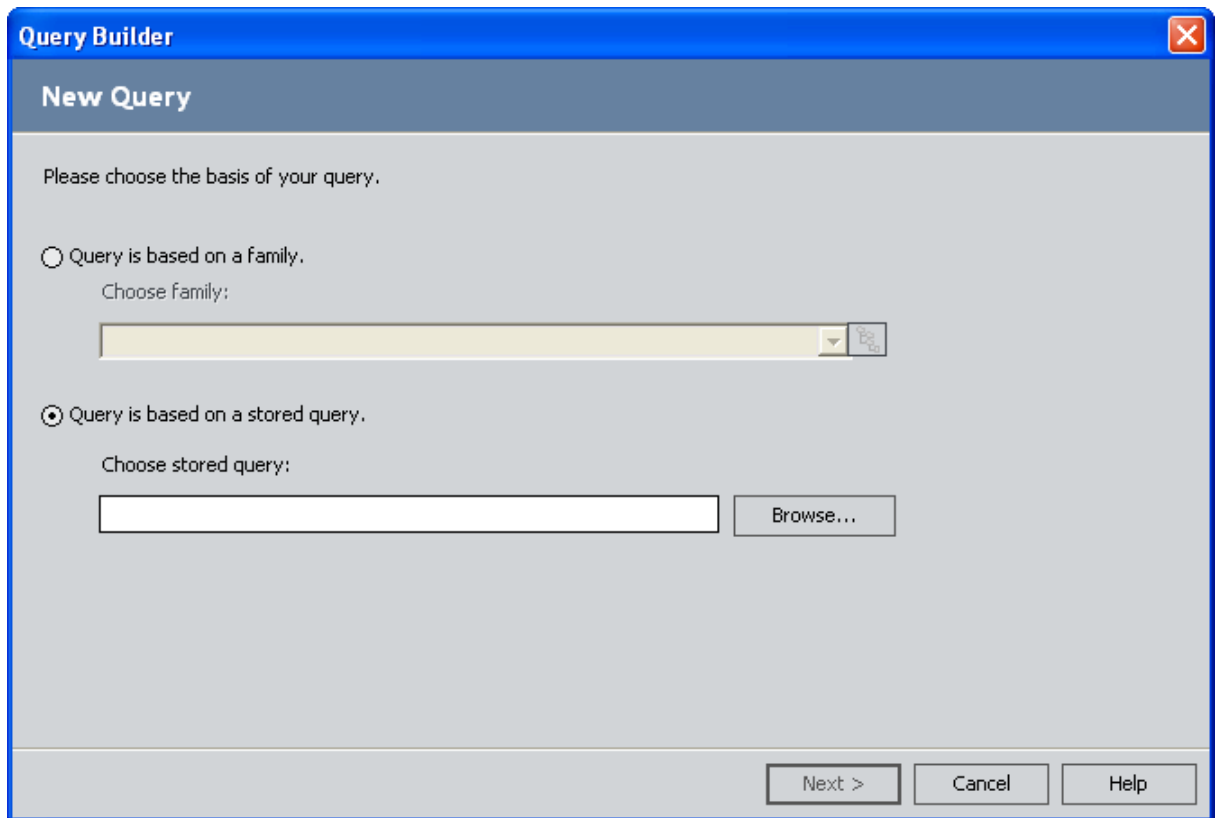
4. Click the **Browse** button.

   The **Load Query** dialog box appears.
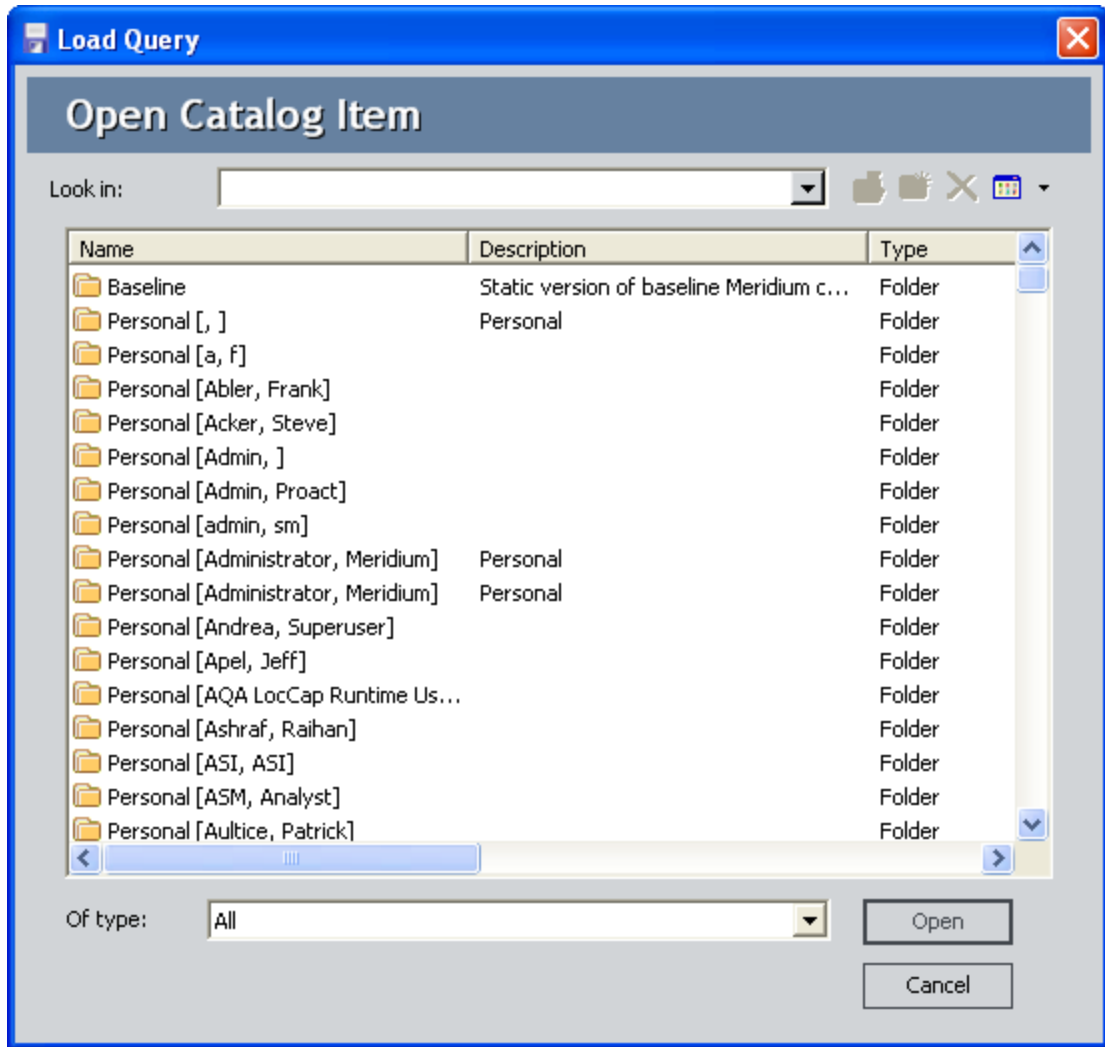
5. In the Catalog, navigate to the query that you want to use, select it, and click the **Open** button.

   The file path to the selected query appears in the **Choose a stored query** text box.
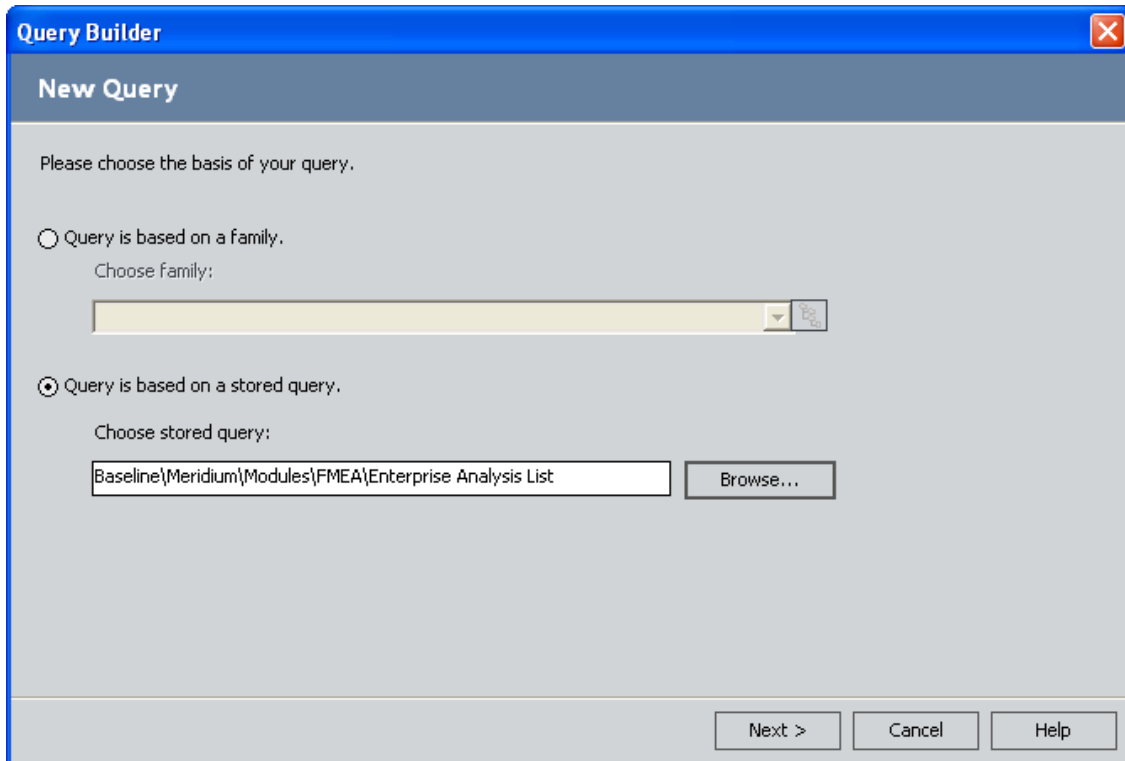
Note: The query that you choose must be a Select query.

6. Click the **Next** button.

The **Query Builder** displays the next step, which shows the families and fields included in the query that you selected.
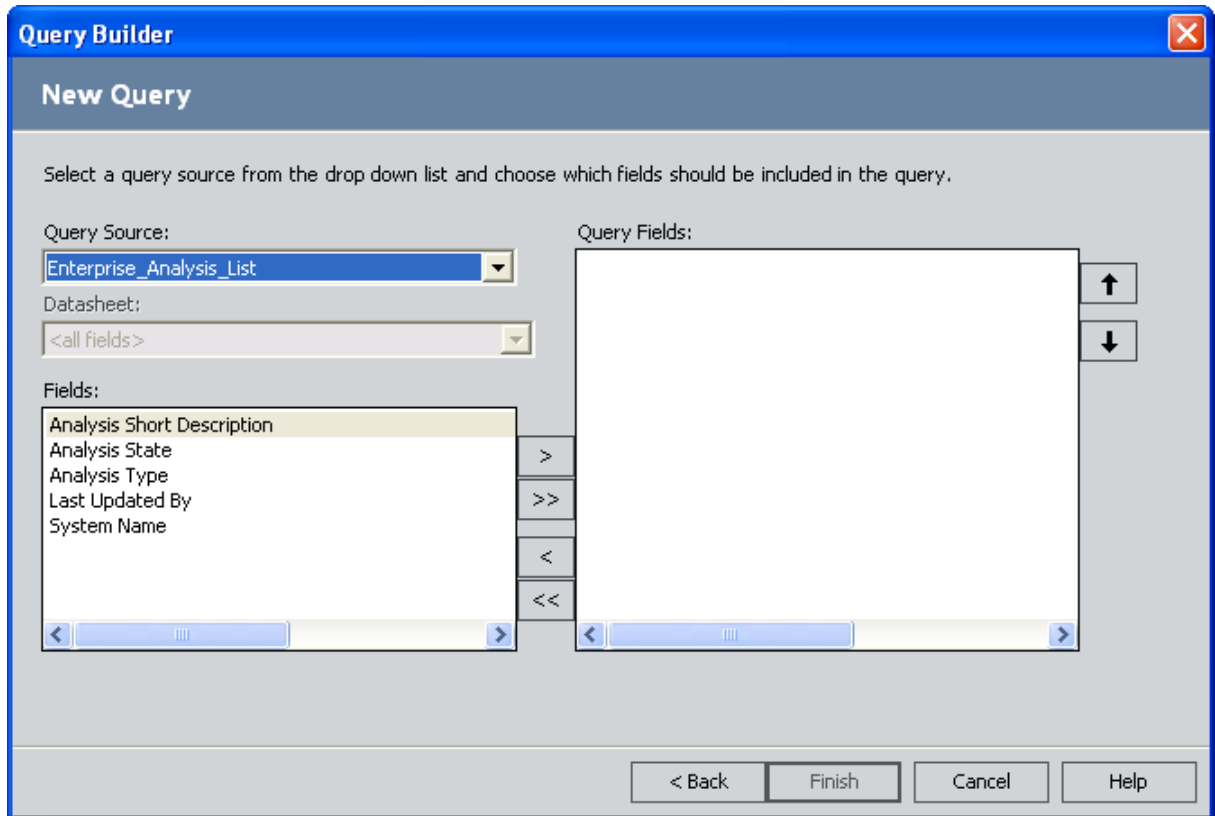
7. Select the fields that you want to include in the new query. To do so, in the **Fields** list, select the desired fields, and click the right arrow button to add them to the **Query Fields** list. You can select multiple fields or click the double right arrow button to add all fields to the query. Similarly, you can use the left arrow and double left arrow buttons to remove fields from the query.

> **Note:** If you click the **Back** button at this point, any field selections that you have made will be lost. If you need to return to a previous screen in the **Query Builder**, you will need to re-add the desired fields to the query.

8. Click the **Finish** button.

   The query design for the new query appears.

9. To run the query, on the **Query Tasks** menu, click the **Run Query** link.

   -or-

   To save the query, on the **Common Tasks** menu, click the **Save Query** link.

Creating Crosstab Queries

To create a Crosstab query:

1. In the Meridium APM Framework, open an existing Select query, or create a new Select query.

2. Access the query design.

3. On the **Query Tasks** menu, click **Options**, and then click **Crosstab Query**.

   The design grid changes to display rows that are appropriate for a Crosstab query.

4. For the field that you want to use as columns in the results grid, in the **Crosstab** cell, select **Column Heading**.

   > **Note:** The **Total** cell that corresponds to the **Column Heading** field must be set to **Group By**.

5. For the field that you want to use as rows in the results grid, in the **Crosstab** cell, select **Row Heading**.

   > **Note:** The **Total** cell that corresponds to at least one **Row Heading** field must be set to **Group By**.

6. For the field that you want to use as the aggregate (i.e., the intersection of the row and column), in the **Crosstab** cell, select **Value**.

   > **Note:** The **Total** cell that corresponds to at least one **Value** field cannot be set to **Group By**, **Expression**, or **Where**. Also, you cannot use as the Value a field where an expression has been defined to return a numeric value. If you create a numeric expression in the **Field** cell for a column in a query and you try to use that field as the **Crosstab** Value, an error appears.

   Note also that if you define criteria (i.e., type a value in the **Criteria** cell) for a field that is selected as the Value, an error appears when you run the query.

7. When you are satisfied with your selections, on the **Query Tasks** menu, click the **Run Query** link to see if the query generates the desired results.

Creating Delete Queries

To create a Delete query to delete records in a family:

1. In the Meridium APM Framework, create or open a Select query whose source represents the family whose records you want to delete.

2. [Access the query design](#).

3. On the **Query Tasks** menu, click **Options**, and then click **Delete Query**.

   If your query contains more than one query source, the **Target Query Source** dialog box appears.

   -or-

   If your query contains only one query source, that source is selected automatically as the target source. You can click the **Target Source** link on the **Query Tasks** menu to modify the default selection.

4. If your query contains more than one source, select the family whose records you want to delete, and click the **Select** button.

   > **Note:** If the query contained more than one query source, the query sources that you did not select are removed automatically from the design canvas.

5. In the query design grid, add the fields whose criteria you want to use for deleting records. For example, if you want to delete Motor records where the Manufacturer field contains the value WESTINGHOUSE, you would need to add the Manufacturer field to the design grid.

6. In the **Criteria** cell for the desired field, specify the criteria for record deletion. Any record that matches the specified criteria will be deleted. For example, if you wanted to delete Motor records where the manufacturer is WESTINGHOUSE, you would create the expression **'WESTINGHOUSE'** in the **Criteria** cell for the Manufacturer field.

7. If you want to delete ALL records that match the query criteria, including those records that are linked to one or more records, on the **Query Tasks** menu, click **Options**, and then click **Delete Force**. If you do not select this option, when you run the Delete query, if any records that match the query criteria are linked to one or more records, an error will appear, and the Delete query will not delete any records.

8. On the **Query Tasks** menu, click the **Run Query** link.

   A message appears, indicating how many records will be deleted by the query.

9. Click the **Yes** button to perform the deletion or the **No** button to stop the deletion.

Creating Update Queries

To create an Update query to modify all the records in a family:

1. In the Meridium APM Framework, create or open a Select query that contains the table whose records you want to update.

2. [Access the query design](#).

3. On the **Query Tasks** menu, click **Options**, and then click **Update Query**.

   If your query contains more than one query source table, the **Target Query Source** dialog box appears.

   If your query contains only one query source, that source is selected automatically as the target source. You can click the **Target Source** link on the **Query Tasks** menu to modify the default selection.

4. In the **Target Query Source** dialog box, choose the target table, family, or link (i.e., the table, family, or link that you want to update), and click the **Select** button.

   > **Note:** If the query design canvas contained more than one query source, the query sources that you did not select are removed automatically from the design canvas.

5. In the design grid, in the **Update To** cell for the fields that you want to modify, type the expression or value that you want to use to update the current value. For example, if you wanted to update Motor records where the manufacturer is WEST so that the manufacturer would be changed to WEST&LONG, you would create the criteria expression `'WEST&LONG'` in the Asset Manufacturer query field for the Motor family.

6. On the **Query Tasks** menu, click the **Run Query** link.

   A message appears, indicating how many records will be updated by the query.

7. Click the **Yes** button to perform the update or the **No** button to stop the query.

   > **Note:** Sometimes, when you run an Update query that will update a very large number of records, an error might appear. If this happens, adjust your query criteria to reduce the number of records that will be updated at one time.

Creating Append Queries

To create an Append query, you will start with a Select query. You can open an existing Select query or create one from scratch to use as the starting point for a new Append query.

To create an Append query:

1. In the Meridium APM Framework, create a Select query or open an existing one that will return the records that you want to add to another family.

2. Access the query design.

3. On the **Query Tasks** menu, click **Options**, and then choose **Append Query**.

   The **Target Query Source** dialog box appears.
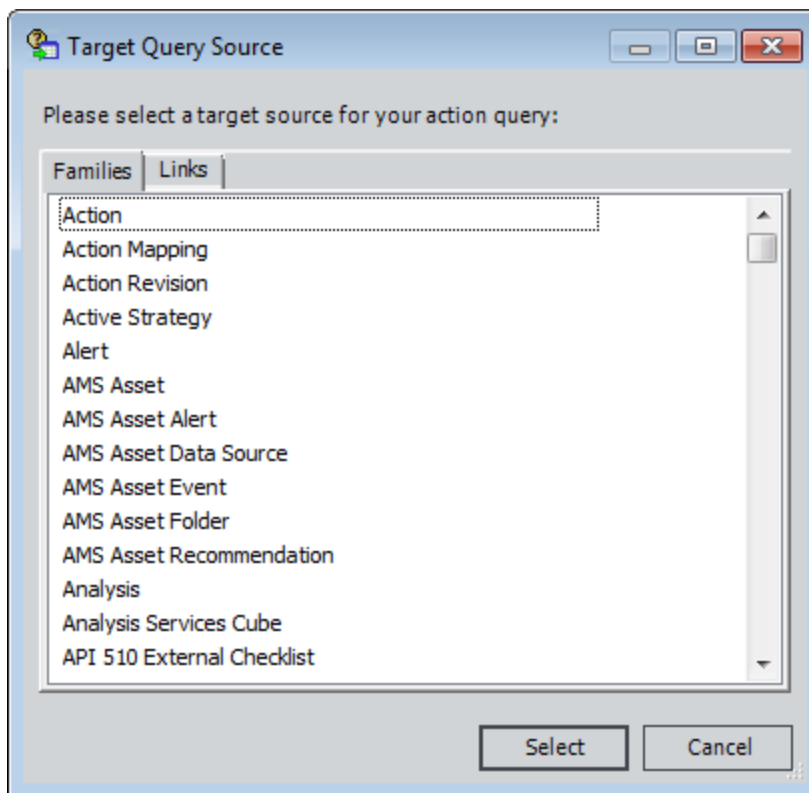
4. In the list of families, select the target query source, which is the family to which records will be appended.

5. Click the **Select** button.

   The **Append Query Target Source** dialog box appears.
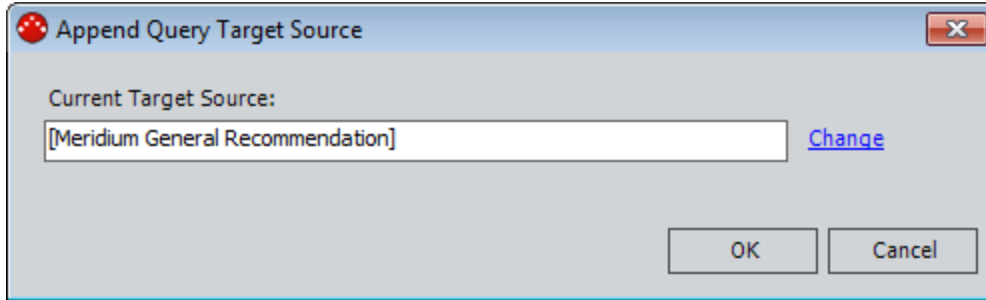
6. Click **OK** to confirm your selection.

   -or-

   Modify your selection by clicking the **Change** link, and then click **OK**.

   > **Note:** After you select the target family, the **Target Source** link will appear on the **Query Tasks** menu. You can click this link to modify the target family if necessary.

   The design grid is updated to include the **Append To** row, where you can define the field mappings for the append operation.
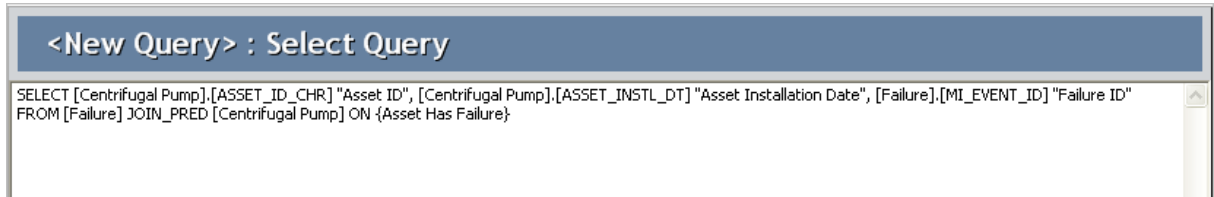
7. In the **Append To** cells, select the field in the target family to which you want to map values from the source family. For example, if you were appending records from one Recommendation family to another Recommendation family, you might map the Recommendation Headline field in the source family to the Recommendation Headline field in the target family.

8. When you are finished, on the **Common Tasks** menu, click the **Save Query** link.

   The Append query is saved.

Accessing the SQL Code

To access the SQL code:

- On the **Query Builder** page, on the **Query Tasks** menu, click the **SQL View** link.

  The SQL code appears.
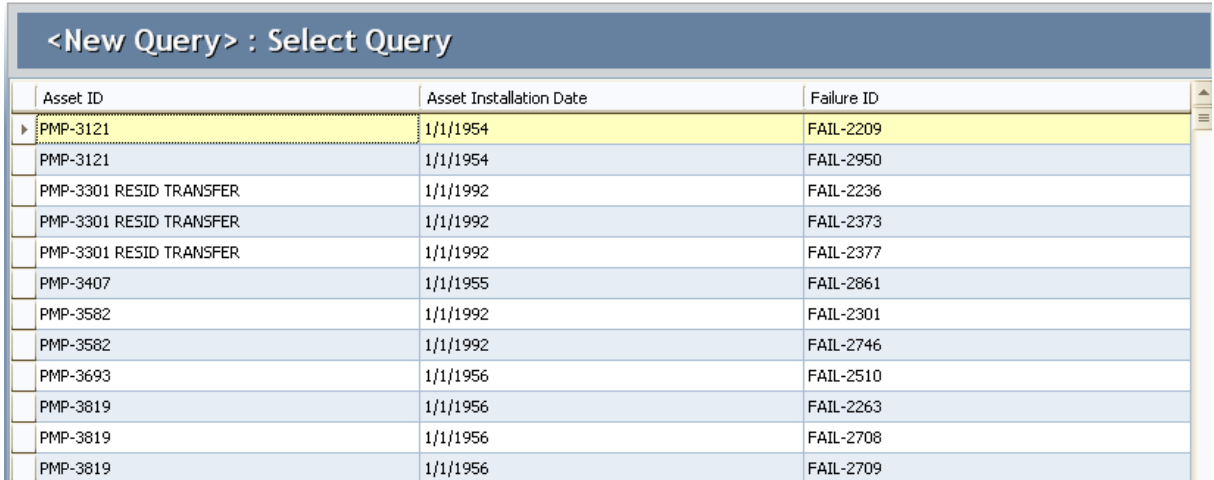
Accessing the Query Results

To access the query results:

- On the **Query Builder** page, on the **Query Tasks** menu, click the **Run Query** link.

  The query results appear.

Asset ID	Asset Installation Date	Failure ID
PMP-3121	1/1/1954	FAIL-2209
PMP-3121	1/1/1954	FAIL-2950
PMP-3301 RESID TRANSFER	1/1/1992	FAIL-2236
PMP-3301 RESID TRANSFER	1/1/1992	FAIL-2373
PMP-3301 RESID TRANSFER	1/1/1992	FAIL-2377
PMP-3407	1/1/1955	FAIL-2861
PMP-3582	1/1/1992	FAIL-2301
PMP-3582	1/1/1992	FAIL-2746
PMP-3693	1/1/1956	FAIL-2510
PMP-3819	1/1/1956	FAIL-2263
PMP-3819	1/1/1956	FAIL-2708
PMP-3819	1/1/1956	FAIL-2709

About Query Results with Prompts

When you run a query that contains one or more prompts, you will see either:

- A dialog box, where you can specify values in each prompt.
- The query results with prompts at the top of the workspace.

You will see the query results with prompts at the top of the workspace only if you open a query from a URL that is configured to display the prompts in this way. You will see the dialog box if you open a query using any other method.

For instance, if you open a query from the Catalog, prompts will appear in a dialog box, which disappears when the results are returned. In the following image, a prompt exists on the Asset Manufacturer field to limit the query results based on the manufacturer. If you were to access this query via the Catalog or a URL that does not specify that the prompt should appear at the top of the results, a dialog box would appear, prompting you to supply the desired value for the manufacturer.

If this same query were accessed via a URL that specifies that the prompt should appear at the top of the results, the prompt would appear at the top of the results. In the following image, the query results are filtered based on the prompt value *CBI*.

When you see a prompt, the prompt caption will most likely guide you to provide an appropriate value.

In some cases, however, prompts can be difficult to use. This is especially true if the prompt does not provide a list of values and you do not know the exact value to type in the prompt. If you see a prompt and you do not know the value that you should type, you can try typing the percent sign (%). If the query was configured to accept this wild-card, the query will return records where the prompt field contains *any* value.

Note that if you see a prompt and you want the query to return records where the prompt field does not contain a value, you can leave the prompt empty and click **OK**. If the query was configured to accept empty prompts, the query will return records where the prompt field does not contain a value.

Exporting a Result Set to a File

You can export your query results to create an external Microsoft Excel, XML, or text file. Note that this export method creates a file that is external to the Meridium APM Framework application. You can also create a file by exporting your query results as a Meridium APM dataset.

To export a query results set:

1. In the Meridium APM Framework, for the query whose results you want to export, access the query results.

2. On the **Other Tasks** menu, click the **Export To File** link.

   If you chose to export a multi-page result set, a message appears, indicating that only the records on the current page will be exported.

3. If you want to export the records on the current page, click **OK**.

   The **Query Results Export Wizard** appears.

4. In the **Data Source** list choose the desired file type:

   • MS Excel

   • Text

   • XML

5. Click the **Finish** button.

   The **Export to File** dialog box appears.

6. Navigate to location to which you want to save the file.

7. In the **File name** text box, type a name for the file.

8. Click the **Save** button.

   The results are saved to the location that you selected.

Creating a Dataset from Query Results

A dataset is a static file of data that is stored in the Catalog. After a dataset exists, you can use it to create reports or generate graphs. You may want to save your query results as a dataset in order to save a "snapshot" of the data and later perform further analysis on the data using other Meridium APM tools.

To create a dataset from a query results set:

1. In the Meridium APM Framework, for the query whose results you want to use to create a dataset, access the query results.

2. On the **Other Tasks** menu, click the **Export To Dataset** link.

   If you chose to export a multi-page result set, a message appears, indicating that only the records on the current page will be exported.

3. If you want to export the records on the current page, click **OK**.

   The **Save Dataset** dialog box appears.

4. In the Catalog, navigate to the place where you want to save the dataset.

5. In the **Name** text box, type a file name for the dataset.

6. If desired, type a description of the data in the **Description** text box.

7. If you want to create a link to the dataset on your Home Page, select the **Add to Homepage** check box.

8. Click the **Save** button.

   The query results are saved as a dataset.

Modifying Queries

To modify an existing query:

1. In the Meridium APM Framework, open an existing query.

   The query runs, and the results appear.

2. To modify the query design, on the **Query Tasks** menu, click the **Design View** link. Then customize the query as desired.

   -or-

   To modify the SQL code directly, on the **Query Tasks** menu, click the **SQL View** link.

3. When you are finished making changes to your query:

   - On the **Query Tasks** menu, click the **Run Query** link to run the query and view the results.

     -or-

   - On the **Common Tasks** menu, click the **Save Query** link to save your changes.

Saving Queries

After you create a query, you can save it so that you or other users can access it again later. Queries are saved as Catalog items to the Meridium APM Catalog. After a query has been saved to the Meridium APM Catalog, whenever you make changes to it, you will need to re-save the query to retain your changes.

Saving a query to the Catalog is similar saving any other item to the Meridium APM Catalog, but consider the following query-specific considerations:

- When you save a query while viewing the query design or the SQL code, any saved personal or Super User grid layout will be discarded. The default layout will be saved and will be used the next time any user runs the query.

- When you save a query while viewing the results, the current grid layout will besaved as a personal layout for the current user. This will override the default layout and any Super User layout defined for the query.

- Links to queries whose names contain special characters may not function correctly. For example, the following issues result from using double quotation marks ("), the number sign (#), or angle brackets (<>):

  - If a query name contains double quotation marks (") and you send a link to the query to an email recipient or the desktop, clicking the email link or the desktop shortcut will open the **New Query** window instead of the saved query.

  - If a query name contains the number sign (#) and you send a link to the query to an email recipient or the desktop, clicking the email link or the desktop shortcut will cause Meridium APM to display a message that the query cannot be loaded.

  - If a query name contains angle brackets (<>) and you send a link to the query to an email recipient, clicking the email link will cause Meridium APM to display a message that the query cannot be loaded. Note that if a query name contains angle brackets and you send a link to the query to the *desktop*, the shortcut will function correctly.

- Depending upon your query privilege setting, you may need to run the query before you can save it.

- To save a Super User grid layout for a query, you must use the Save for All Users option on the shortcut menu in the results. The save options on the **Common Tasks** menu will not save the current layout for all users, even if you are logged in as a Super User at the time you save the query.

To save a query to the Meridium APM Catalog:

1. While viewing the query on the **Query Builder** page, on the **Common Tasks** menu, click the **Save Query** link.

   If the query has already been saved, any changes that you have made will be saved to the current Catalog folder and with the existing Catalog item properties. If the query has not yet been saved, the **Save Query** dialog box will appear, as shown in

the following image. The following instructions assume that you are saving the query for the first time.
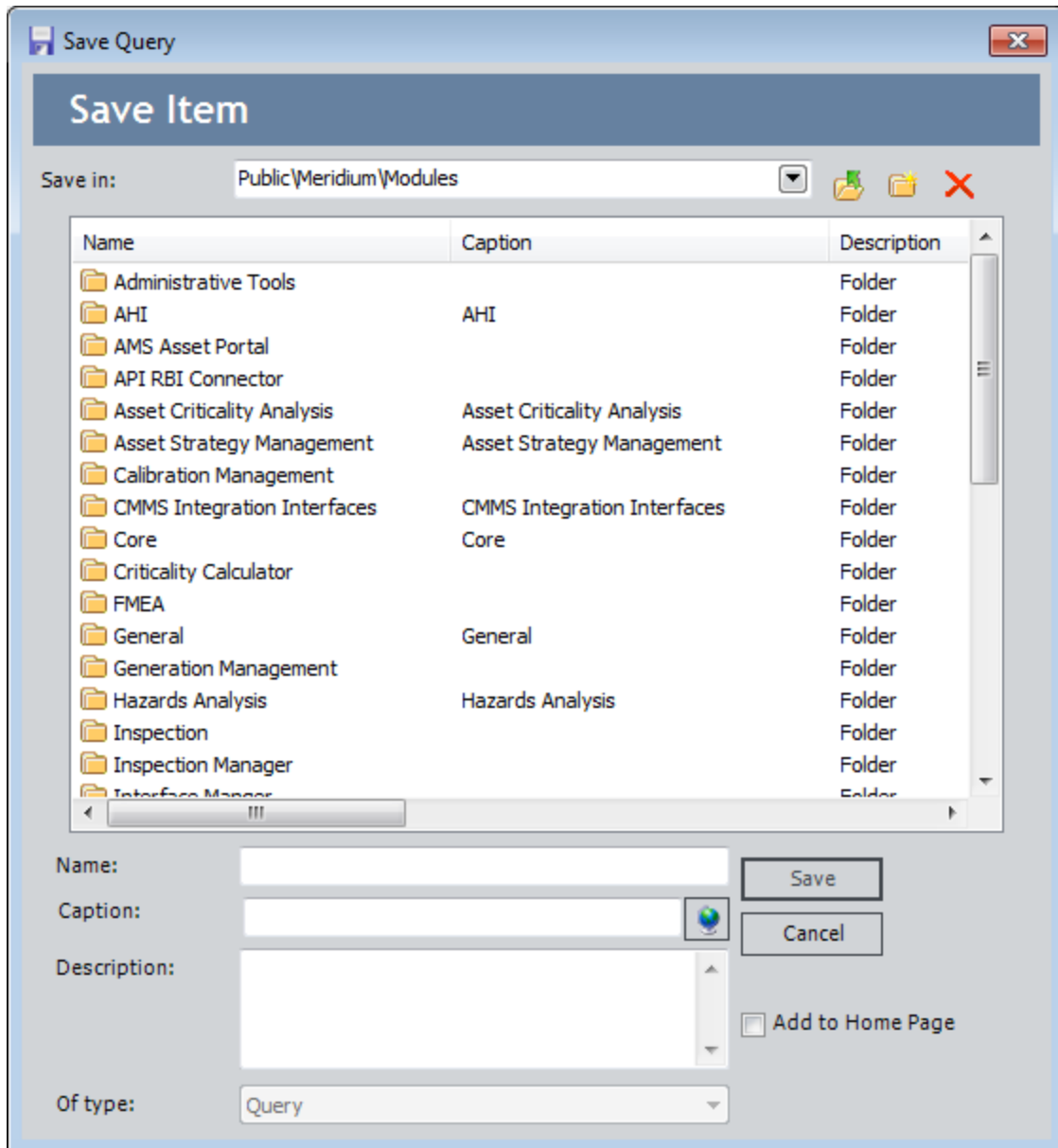
Note: If you are viewing a saved query on the **Query Builder** page and you click the **Save As** link on the **Common Tasks** menu, the **Save Query As** dialog box will appear so that you can save a *copy* of the current query with a different name or to a different location. You can use the following instructions to complete that task, as the process is the same as saving a query for the first time.

2. Navigate to the Catalog folder in which you want to save the query.

3. In the **Name** text box, type a name for the query. The name is required and must be unique to the Catalog folder in which you are saving the query.

4. The **Caption** text box is populated automatically with the value that you typed in the **Name** text box.

5. If desired, click the 🌐 button to manage translations for that string.

6. If desired, type a description of the dashboard in the **Description** text box. This is not required to save the query.

7. If desired, select the **Add to Home Page** check box if you want to add a link to the query to a Home Page. This is not required to save the query.

8. Click the **Save** button.

    If you:

-   Selected the **Add to Home Page** check box, the query is saved to the Catalog, and the Send to Home Page - Select Home Page dialog box is displayed.

    Depending upon your membership in the MI Power User Role Security Group and whether or not you are a Super User, you can use the **Send to Home Page - Select Home Page** dialog box provide a link to the dashboard to your Personal Home Page, a Group Home Page, or the Personal Home Page for a particular Security User.

- Did not select the **Add to Home Page** check box, or you have completed the process of adding the dashboard to a Home Page, the **Save Query** dialog box closes, and the query is saved to the Catalog.

Deleting Queries

To delete a saved query:

1. In the Meridium APM Framework, for the query that you want to delete, [access the query design](#).

2. On the **Common Tasks** menu, click the **Delete Query** link.

   A confirmation message appears, asking if you really want to delete the query.

3. Click the **Yes** button to delete the query from the Catalog.

Accessing the Query Design

To access the query design:

- On the **Query Builder** page, on the **Query Tasks** menu, click the **Design View** link.

  The query design appears.
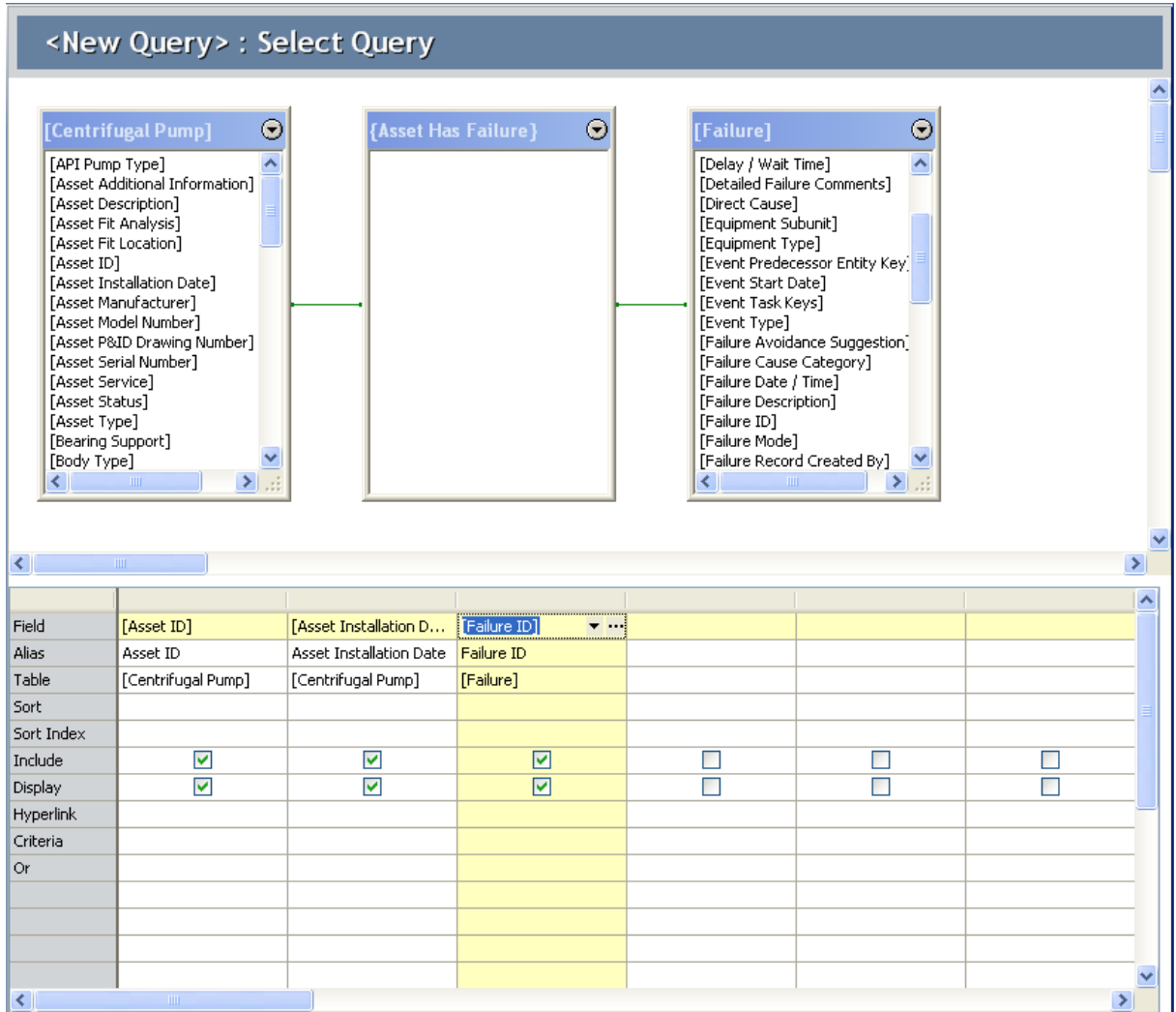
Adding Entity Families, Relationship Families, and Queries

The following instructions provide details on adding entity families, relationship families, or queries as query sources by selecting them individually. If you add as query sources both a relationship family and the associated entity families that participate in a relationship with one another, a link will be created automatically between those related query sources. To add related families as query sources in this way, you must be familiar with the relationships that exist in your database.

> Œ Hint: Alternatively, you can add related families as query sources using the Add Related Family option, which lets you select the related families from the list of relationships that are defined for a given entity family in your database.

To add an entity family, relationship family, or query as a query source:

1. In the Meridium APM Framework, access the query design.

2. On the Query Tasks menu, click the Add Source link.

    The Add Query Source dialog box appears.
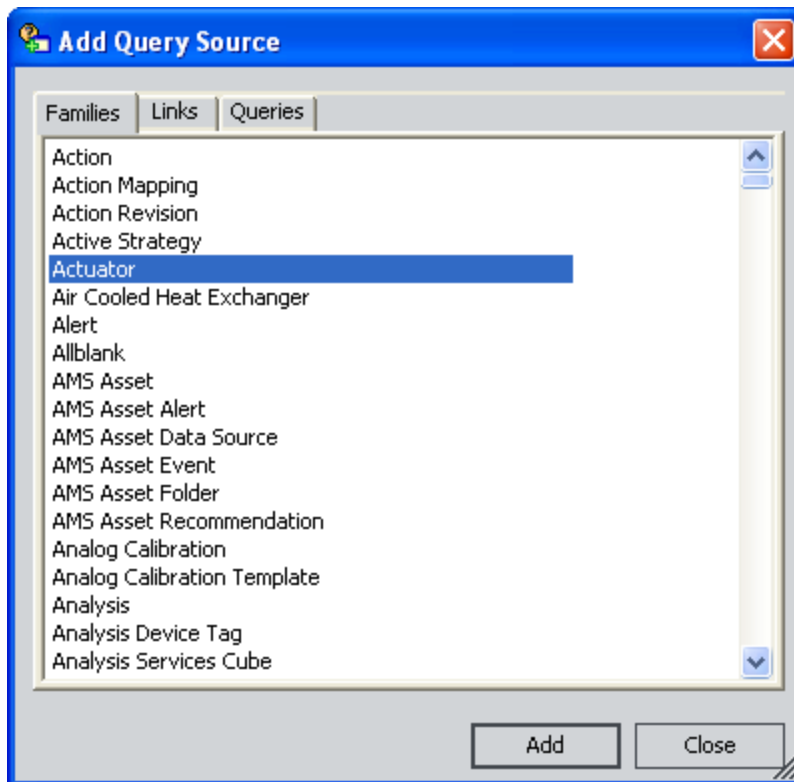
3. Click the following tabs as needed, select the desired element(s), and then click the Add button:

- **Families:** Displays the list of entity families to which you have been granted at least *View* privileges in the Configuration Manager.

- **Links:** Displays the relationship families to which you have been granted at least *View* privileges in the Configuration Manager. Note that if you add multiple entity families that are related through the same relationship, you will need to add the relationship family multiple times (i.e., once for *each* predecessor family in the selected relationship). For example, if the Pump, Analyzer, and Compressor families are all related to the Failure family through the Has Failure relationship family, you would need to add the Has Failure relationship family to the query as a successor of each family. In other words, you would need to add the Has Failure relationship family three times.

- **Queries:** Displays the list of Catalog folders to which you have at least *View* privileges.

  As you click the **Add** button, your selections will be displayed on the design canvas. The **Add Query Source** dialog box will remain open until you click the **Close** button so that you can make as many selections as you need.

4. When you have finished adding query sources, click the **Close** button.

   The design canvas returns to focus.

> **Note:** If you add one or more query sources that participate in the *same* relationship with another query source that was already part of the query, when you add the additional query sources, a message will appear, indicating that you must link the query sources manually. This is because the Meridium APM system is unable to determine automatically which link you want to create in these cases. For example, suppose you add the families Site, Area, Unit, and Location Contains Sub-Location *in that order*, where you added the Location Contains Sub-Location relationship family after closing the **Add Query Source** window and opening it again. If the Location Contains Sub-Location family can be used to relate the Site family to the Area family *and* the Unit family to the Area family, a message will appear after you add the Location Contains Sub-Location family, indicating that you must manually create the link between the appropriate families. You can do so using the **Link With** option, which is accessible via the ⊙ button in the query source lists.

Adding Related Families

The following instructions provide details on adding as a query source any entity family that participates in a relationship with an existing query source, as well as the relationship family that relates the two entity families. When you do so, the selected entity family and relationship family will be added to the design canvas as query sources and linked automatically to the existing query source. This method for adding families as query sources is useful if you are not familiar with the relationships that are defined in your database.

> Œ Hint: Alternatively, if you are familiar with the relationships that are defined in your database, you can add the related entity and relationship families individually using the Add Source option.

To add a related family and corresponding relationship family as query sources:

1. In the Meridium APM Framework, for the query to which you want to add a related family and corresponding relationship family as query sources, access the query design.

2. On the design canvas, in the query source list for the family whose related family you want to add as a query source, click the ⊙ button in the upper, right corner.

   A menu appears.

3. Click the **Add Related Family** option.

   The **Add Related Family to [<query source>]** window appears, displaying the list of families that participate in a relationship with the selected query source, as well as the corresponding relationship family. The following image shows an example of what this window looks like when you access it via the *Equipment* family.
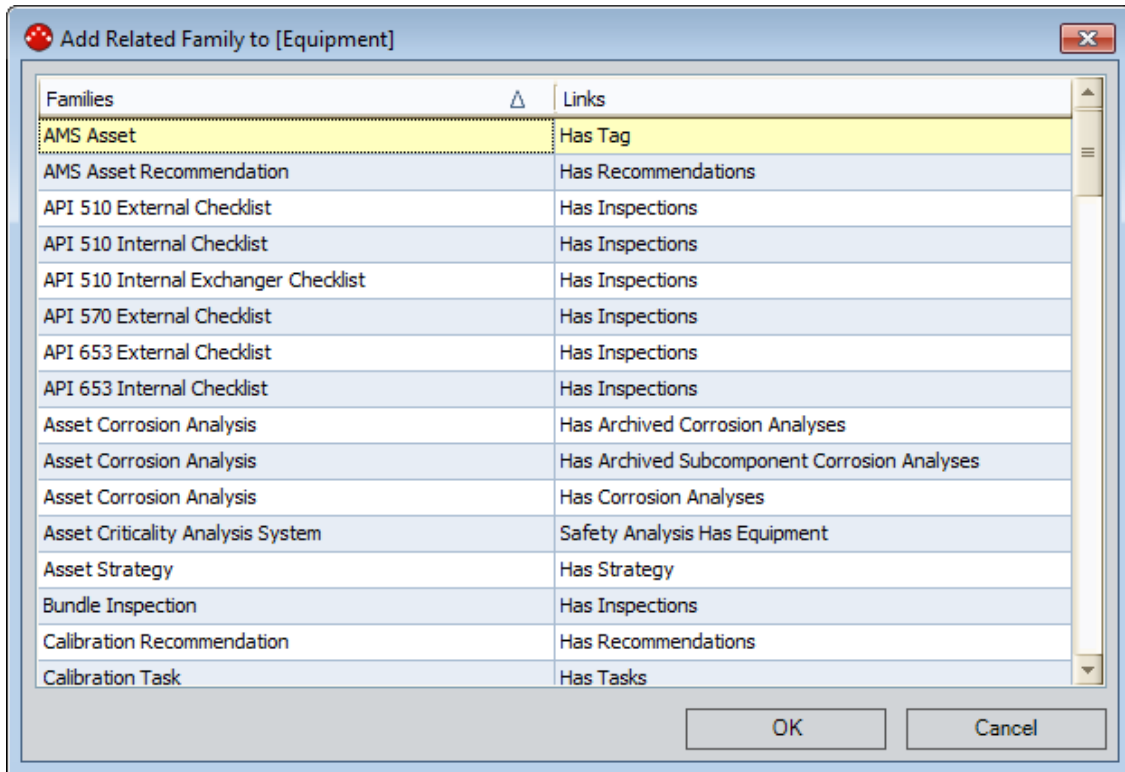
Note: The **Add Related Families** option is enabled only if the selected query source is an entity family that participates in at least one relationship.

4. Select the row containing the related entity family and the corresponding relationship family that you want to add to the design canvas, and then click **OK**.

The selected entity family and relationship family are displayed on the design canvas.

Œ Hint: You can select multiple rows to add multiple families and relationships at one time.

The following image shows an example of what the design canvas looks like after two sets of related families are added as query sources that are linked to the *Equipment* family.
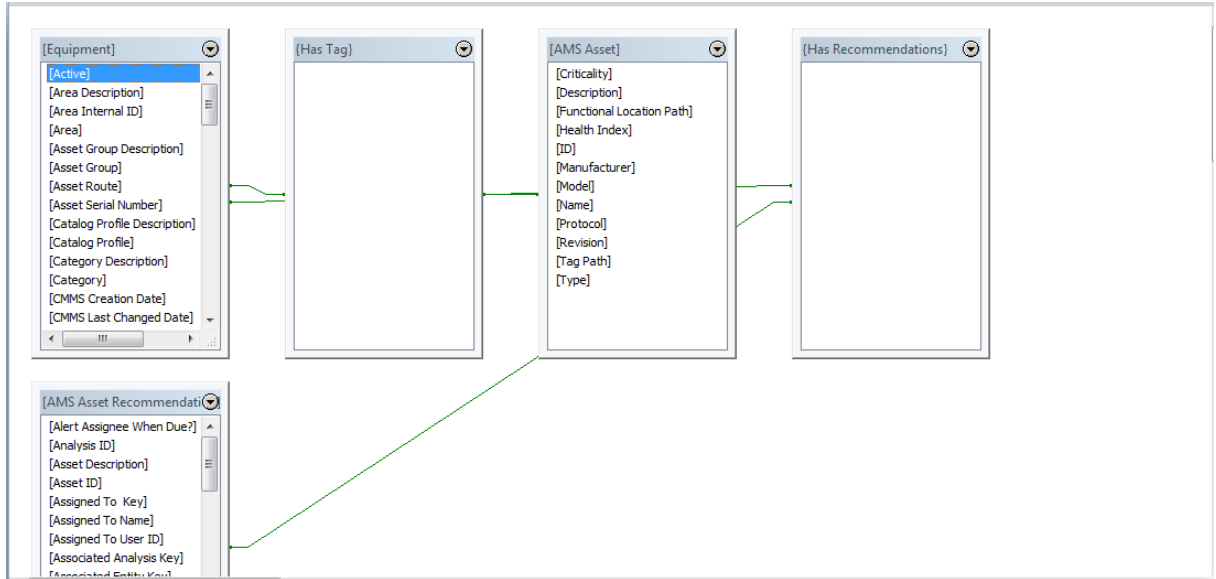
Defining the Alias for a Query Source

The alias for a query source is automatically set as the family caption when you add the source, but you can change the alias. You would want to do this, for example, if your query is configured as described in the following list:

- The query contains the same query source multiple times.
- You are adding a hyperlink to the query.
- You are passing in to the hyperlink a System Code field (from one of the duplicate query sources) that is not included in the query results.

The following instructions assume that you have added at least one query source.

To define an alias for a query source:

1. In the Meridium APM Framework, for the query containing the source whose alias you want to modify, access the query design.

2. On the design canvas, in the query source list for the source whose alias you want to modify, click the ⊙ button in the upper, right corner.

3. On the submenu that appears, click the **Properties** option.

   The **Query Source Properties** dialog box appears.
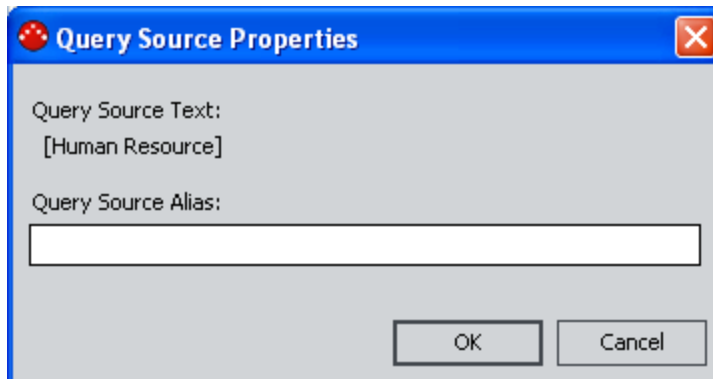
3. In the **Query Source Alias** text box, type the name you want to use for the query source, and click **OK**.

   The query design is updated to reflect the new query source alias. In the following image, the updated alias is *New Alias*.

New_Alias

[Address1]
[Address2]
[Area of Responsibility]
[Available?]
[Badge Id]
[Business Unit]
[City]
[Comments]
[Company]
[Country]
[Culture]
[Department]
[Domain]
[Email Address]
[Facility]
[Fax]
[First Name]
[Job Title]

Field	ENTY_KEY	[Last Name]	[State]	[City]
Alias	ENTY_KEY	Last Name	State	City
Table	New_Alias	New_Alias	New_Alias	New_Alias
Sort				
Sort Index				
Append To				
Criteria			((? :s :caption='Selec...	((? :s :caption='Selec...
Or				

About Joining Fields in Query Sources

In many cases, you will want to query the database for information that exists in more than one family. When you do so, you may also want to join at least one field from each family with a similar field in another family for the purpose of establishing an ad hoc association between the two families. By joining a field in one family to a field in another family, you create a query *join*. There are two types of query joins:

- **Inner Join:** Returns a row for every record where the values in the joined fields from both families are equal. An inner join returns only those records that satisfy the join conditions, so any unmatched records are dropped from the result set.

- **Outer Join:** Returns a row for every record where the values in the joined fields from both families are equal and every record from one of the families where the value in the joined field is not equal to the value in the joined field in the other family. An outer join can be either a *left outer join* or a *right outer join*.

When you add query sources to the design canvas and join their fields, the family you added first is referred to as Table #1, and the family you added second is referred to as Table #2.

A *left outer join* returns all records from Table #1, regardless of whether they are related to an entity from Table #2. The results also include only records from Table #2 where the value in the joined field is equal to the value in the joined field in a record in Table #1.

A *right outer join* returns all records from Table #2, regardless of whether they are related to records in Table #1. The results also include only records from Table #1 where the value in the joined field is equal to the value in the joined field in a record from Table #2.

Viewing the Properties of an Existing Join

To view the properties of an existing join:

1. In the Meridium APM Framework, for the query containing the join whose prop-erties you want to view, access the query design.

2. On the design canvas, click the line that connects two query sources, and click **Link Properties**.

   The **Link Properties** window appears.
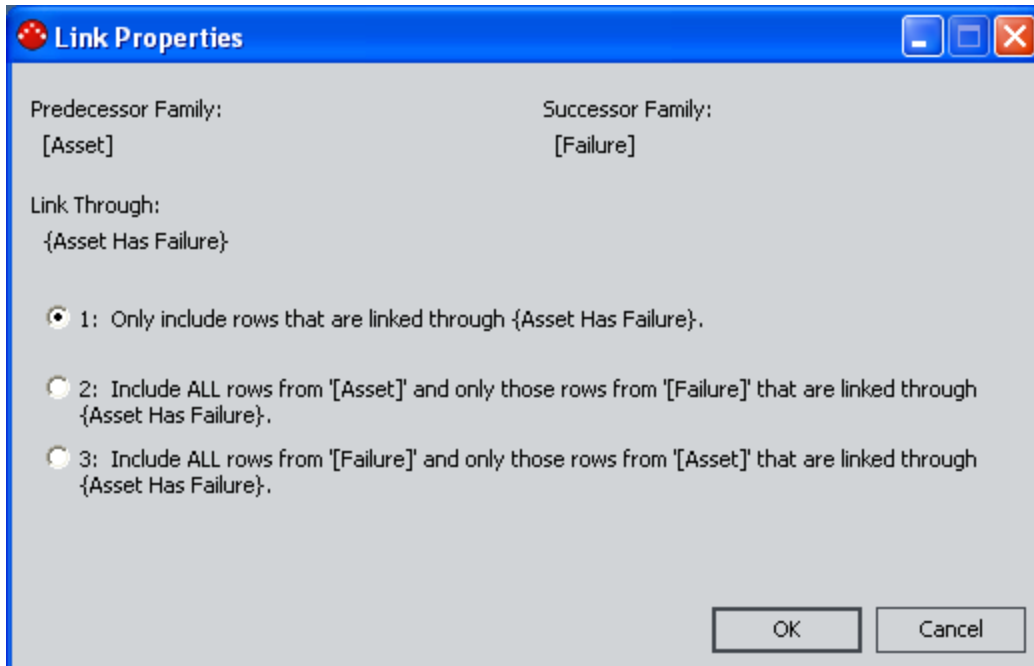
From the **Link Properties** window, you can determine the query source families, indicated by the labels **Table #1** and **Table #2**. You can also determine the joined fields, which are listed in the **Field #1** and **Field #2** lists. In addition, you can determine if the join is an inner join, a left join, or a right join. An inner join is rep-resented by the first option, a left join is represented by the second option, and a right join is represented by the third option.

About Viewing Only Matching Records

When you use as query sources two related entity families and the relationship family that relates them, the Meridium APM system creates an inner join between the two entity families by default. This inner join causes the query results to include only the records that are linked through the specified relationship.

For example, suppose you use as query sources the *Equipment* entity family, *Work History* entity family, and *Has Work History* relationship family. Assuming that the Has Work History family relates the Equipment family to the Work History family, the Meridium APM system will create an inner join between the Equipment and Work History families by default. As a result, the query results will return only Equipment records that are linked to a Work History record through the Has Work History relationship. The query results will not include any Equipment records that are not linked to a Work History record or any Work History records that are note linked to an Equipment record.

If two entity families are not related through a relationship family, you can create a manual, inner join to connect one or more fields in one family to one or more fields in the other family.

For example, suppose you notice that two pieces of equipment, such as a centrifugal pump and an air cooled heat exchanger, have been malfunctioning recently. You suspect that the failures could be caused by human error when installing the machinery. You decide to query the database to find instances where these pieces of equipment were installed on the same day by the same person.

To do so, you would need to add as query sources the Centrifugal Pump family and the Air Cooled Heat Exchanger family and join the Asset Installation Date fields and the Responsible Installer fields between the two families via an inner join. Doing so would return only records where a Centrifugal Pump was installed by the same person on the same date as an Air Cooled Heat Exchanger.

Note that the join is also part of the SQL code.

Creating a Manual Join to Return Only Matching Records

To create an inner join to return only records that satisfy the join conditions:

1. In the Meridium APM Framework, for the query in which you wan to create a manual join, access the query design.

2. Add the appropriate query sources.

3. Identify the field on which to create the join. For example, if you want to view records from two entity families where the asset installation date is the same, you would use the Asset Installation Date field of each entity family.

4. Click the field in one query source list and drag it on top of the appropriate field in another query source list.

   The **Custom Link Properties** window appears.
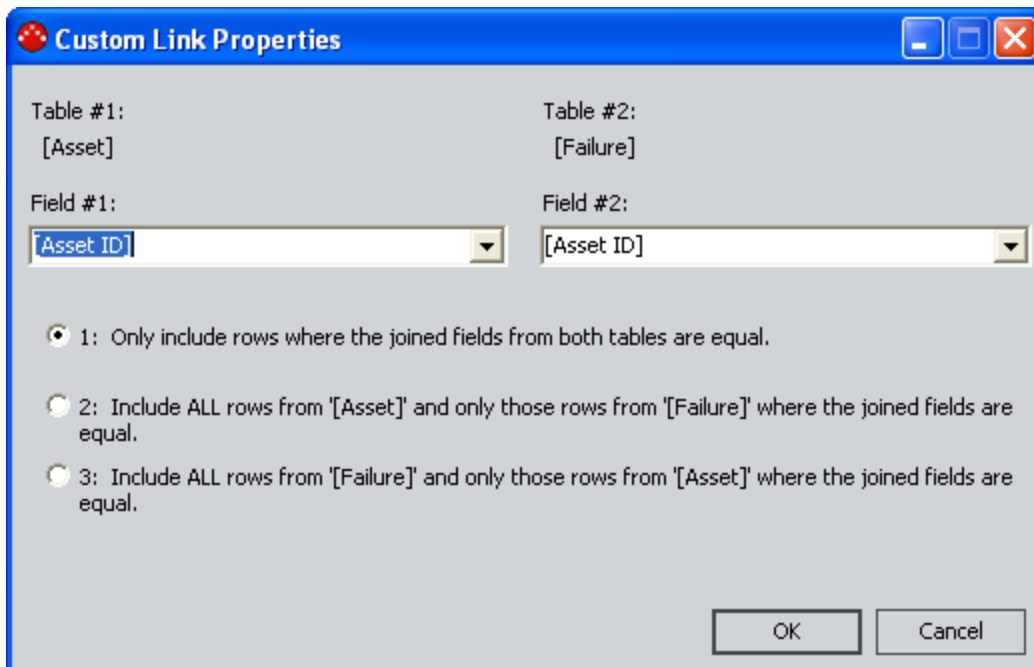
The names of Table #1 and Table #2 appear at the top of the dialog box. These table names represent the names of the selected query source families. The names of the join fields appear beneath the names of the families in the lists labeled **Field #1** and **Field #2**. You can modify the selected fields by choosing a different selection from the **Field #1** list or **Field #2** list.

5. Accept the default selection, **Only include rows where the joined fields from both tables are equal**. This indicates that the query results will include records from Family #1 only if the value in the joined field matches the value in the joined field

of Family #2. For example, if you are joining a Shell and Tube Heat Exchanger family (Table #1) to a Centrifugal Pump family (Table #2) using the Asset Installation Date field, the results will include only the Shell and Tube Heat Exchanger records that have the same installation date as a Centrifugal Pump record.

6. Click **OK**.

   The custom join is saved with the query. You can view the join properties or delete the join by clicking the line that connects the query sources and selecting the appropriate option from the shortcut menu.

About Viewing Matching and Non-Matching Records

As opposed to an [inner join](#), which returns only records that match according to the join condition and drops unmatched records from the query results, an outer join allows you to return records that satisfy the join conditions and records from one family for which there are no matching records in the other family. You can create a manual outer join for families that are not related through a relationship family, or you can modify the default inner join to create a manual outer join for families that are related through a relationship family.

For example, if you use as query sources two unrelated families, such as the Centrifugal Pump family and the Rotary Pump family, you can create an outer join to view all Centrifugal Pumps and Rotating Pumps whose manufacturer is the same, and all Centrifugal Pumps with a different manufacturer. If the Centrifugal Pump family was added as a query source as Table #1 and the Rotary Pump family was added as Table #2, this would create a left join, which would be indicated in the SQL code.

On the other hand, if you wanted to view all Rotary Pumps and Centrifugal Pumps whose manufacturer is the same, and all Rotary Pumps with a different manufacturer, you would create a right join, which would be indicated in the SQL code.

You can also modify a relationship-driven inner join to create an outer join. For example, suppose that you use as query sources the Centrifugal Pump entity family, the Failure entity family, and the Asset Has Failure relationship family. If your system is configured such that the Asset Has Failure family relates the Centrifugal Pump family to the Failure family, the Meridium APM system automatically creates an inner join between the two entity families.

If you ran the query using the default join, the results would include all Centrifugal Pump records that are linked to a Failure record. If you wanted to modify those results, however, to view all Centrifugal Pumps with their linked failure records and all Centrifugal Pumps that did not have linked failure records, you would need to modify the inner join to create an outer join. This would return Centrifugal Pumps with and without linked Failures, but not Failures without linked Centrifugal Pumps. For example, the following image shows all the Centrifugal Pump records, only three of which have associated Failure records.

Asset ID	Failure ID
PMP-8110	
PMP-8062	
PMP-4943	
PMP-8035	
AQA2 PMP-8281	
PMP-7875	
PMP-4029	
PMP-7872	
PMP-4024	
PMP-7090	
Pump-G-1	
PMP-7831	
PMP-2345	
PMP-6010	FAIL-2108
PMP-4499	FAIL-2109
PMP-4917	FAIL-2110

This type of join is considered a left join, as indicated by the SQL code.

On the other hand, you could also decide to return all Failure records with their linked Centrifugal Pump records and all Failure records that do not have linked Centrifugal Pump records. The results would then look like those shown in the following image.

Asset ID ▽	Failure ID
PMP-3121	FAIL-2209
PMP-1001	FAIL-2718
PMP-1001	FAIL-2313
PMP-1001	FAIL-2240
PMP-1001	FAIL-2135
FF-5542	FAIL-2747
FF-5542	FAIL-2458
FF-5542	FAIL-2411
	FAIL - 01764
	FAIL - 01574
	FAIL - 01570
	FAIL - 01569
	FAIL - 01568
	FAIL - 01567
	FAIL - 01566

From these results, you can see that eight failures are linked to pump and the additional failures that are linked to some other type of equipment.

This type of join is considered a right join, as indicated by the following SQL code:

SELECT [Centrifugal Pump].[ASSET_ID_CHR] "Asset ID", [Failure].[EFAIL_ASSETID_CHR] "Failure ID"
FROM [Centrifugal Pump]**RIGHT JOIN**_SUCC [Failure] ON {Asset Has Failure}

Creating a Join to Return Matching and Non-Matching Records

To create a join to return matching and non-matching records:

1. In the Meridium APM Framework, for the query in which you want to create a join to return matching and non-matching records, access the query design.

2. Add the appropriate query sources.

3. Identify the field that you want to use as the join field.

4. Click the field in one query source list and drag it on top of the appropriate field in another query source list.

   The **Custom Link Properties** window appears.

5. Select one of the following option buttons:

   - **Include all rows from [Table #1] and only those rows from [Table #2] where the joined fields are equal:** The results will include all records from Table #1, regardless of whether they are related to an entity from Table #2. The results will also include only records from Table #2 where the value in the joined field is equal to the value in the joined field in a record from Table #1. This option will create a left join.

   - **Include all rows from [Table #2] and only those rows from [Table #1] where the joined fields are equal:** The results will include all records from Table #2, regardless of whether they are related to records in Table #1. The results will also include only records from Table #1 where the value in the joined field is equal to the value in the joined field in a record from Table #2. This option will create a right join.

6. Click **OK**.

   The custom join is saved with the query.

Modifying an Existing Inner Join to Return Matching and Non-Matching Records

To modify an inner join to create an outer join that will return matching and non-matching records:

1. In the Meridium APM Framework, for the query in which you want to modify an inner join, access the query design.

2. On the design canvas, click the existing join, and click **Link Properties**.

   The **Custom Link Properties** window appears.

2. Select one of the following option buttons:

   - **Include all rows from '[Table #1]' and only those rows from '[Table #2]' that are linked through {[Relationship family]}:** The results will return all records from Table #1 and only those records from Table #2 that are linked to records in Table #1 through the relationship family. This option will create a left join.

   - **Include all rows from '[Table #2]' and only those rows from '[Table #1]' that are linked through {[Relationship Family]}:** The results will return all records from Table #2 and only those records from Table #1 that are linked to records from Table #2 through the relationship family. This option will create a right join.

3. Click **OK**.

   The custom join is saved with the query.

Deleting a Join Between Two Query Sources

To delete a join between two query sources:

1. In the Meridium APM Framework, for the query from which you want to delete a join, access the query design.

2. On the design canvas, click the join connecting two query sources, and click **Delete**.
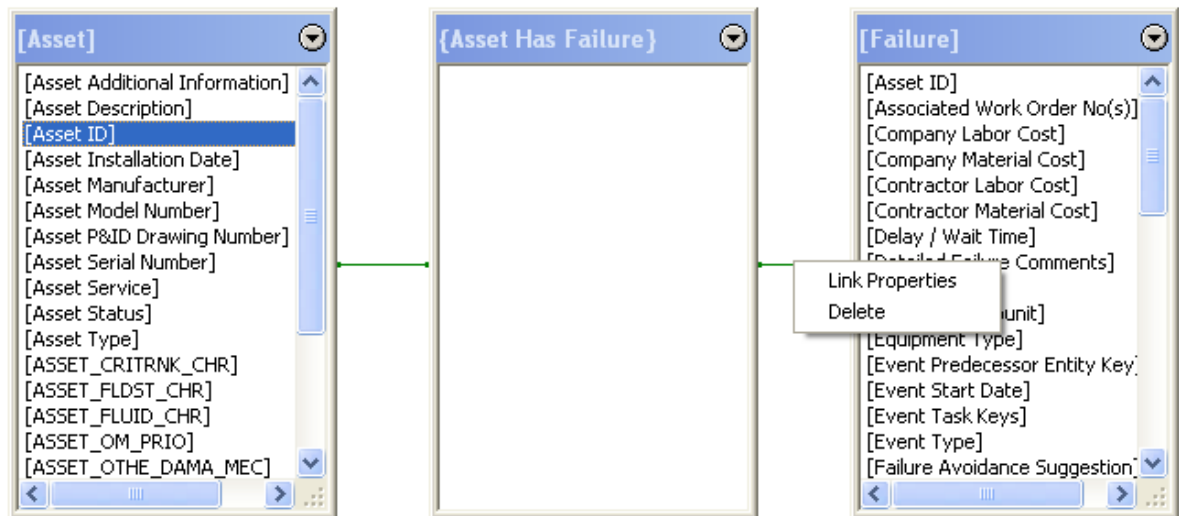
The join is removed from the design canvas. Note that if you delete an inner join between two families that are related through a relationship family, the join and the relationship family are removed from the design canvas.

Displaying System Fields for a Query Source

By default, when you add a query source, system fields for the source are not displayed in the field list. You may, however, want to add system fields to your query design grid.

To display system fields for a query source:

1. In the Meridium APM Framework, for the query containing the source whose system fields you want to view, [access the query design](#).

2. On the design canvas, in the query source whose system fields you want to view, click the ⊙ button in the upper, right corner.

3. Point to **Show Fields**, and then click **Show System Fields**.

    The system fields are appended to the fields in the query source window. System fields are shown on bold and are identified by their field IDs.
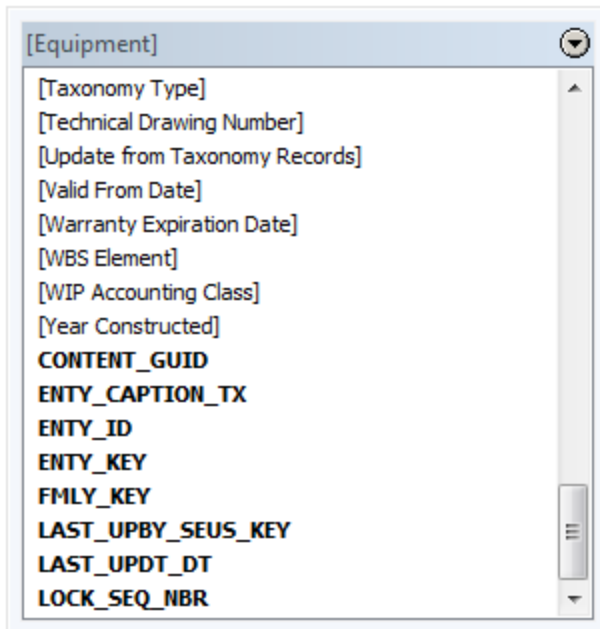
Displaying Inactive Fields for a Query Source

By default, when you add a query source, inactive fields that exist in that family are not displayed in the source list. You may, however, want to add inactive fields to your query design grid. You can male inactive fields available for use in your queries using the following instructions.

To display the inactive fields that exist for a query source:

1. In the Meridium APM Framework, for the query containing a source for which you want to display inactive fields, access the query design.

2. On the design canvas, in the source for which you wan to display inactive fields, click the ⊙ button in the upper, right corner.

3. Point to **Show Fields**, and then click **Show Inactive Fields**.

   The system fields are appended to the fields in the query source window. The field captions are shown in bold, in square brackets. In the following image, the *Asset ID* field is inactive.

   [Actuator]
   [Effective Diaphram/Piston Area]
   [Fail Position]
   [Maximum Design Pressure]
   [Minimum Required Pressure]
   [Shutdown Req. for Maintenance?]
   [Spared ?]
   CONTENT_GUID
   ENTY_CAPTION_TX
   ENTY_ID
   ENTY_KEY
   FMLY_KEY
   LAST_UPBY_SEUS_KEY
   LAST_UPDT_DT
   LOCK_SEQ_NBR
   [Asset ID]
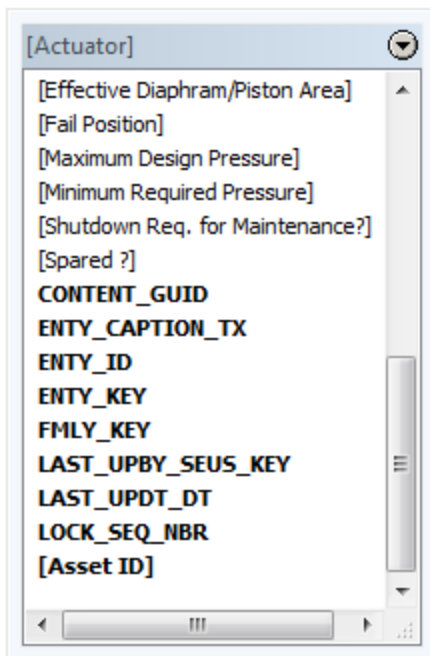
Removing Query Sources

To remove a query source:

1. In the Meridium APM Framework, for the query from which you want to remove a source, access the query design.

2. On the design canvas, in the source that you want to remove, click the ⊙ button in the upper, right corner.

3. On the submenu that appears, click the **Remove** option.

   The query source is removed from the design canvas. Note that if your design canvas contains two related families and the relationship that connects them, when you remove the predecessor or successor family from the design canvas, the relationship family will also be removed. In the following example, if you removed the Air Cooled Heat Exchanger or Failure source, the Asset Has Failure source would also be removed.
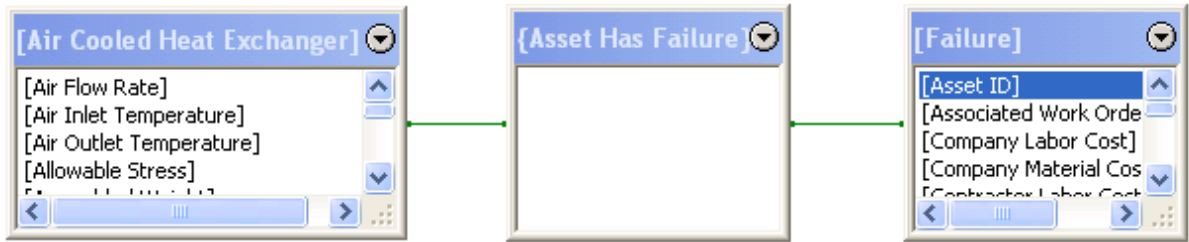
Defining an Alias for a Column

The alias for a field specifies how it will be labeled in the query results and in any report or graph that is created from the query. When you add fields to the design grid via the **Query Builder** or the drag-and-drop method, the alias is set automatically to the field caption. You can modify the alias if desired.

For example, in the following image, for the [Asset ID], [Asset Description], and [Asset Installation Date] columns, the aliases are *ID*, *Description*, and *Installation Date* respectively.

Field	[Asset ID]	[Asset Description]	[Asset Installation Date]
Alias	ID	Description	Installation Date

The aliases will appear as the column headers in the results, as shown in the following image.

\<New Query\> : Select Query

ID	Description	Installation Date	▽
100102	Chiller Plant Cooling Tower Cell #2	6/2/1997	
100101	Chiller Plant Cooling Tower Cell #1	6/2/1997	

When defining aliases, keep in mind the following considerations:

- Within a given query, each alias must be unique. If you specify an alias that has already been defined for another column in the same query, a number will be appended to the end of each duplicate alias to distinguish them from the other aliases in the query. For example, if you specify *Asset* as the alias in four columns, the aliases would be changed to:
  - Asset
  - Asset0
  - Asset1
  - Asset2

  Each duplicate alias will be numbered in this way using the next available number (i.e., 3, 4, 5, and so on).

  > **Note:** Automatic numbering of duplicate aliases occurs only for default alias values. Translated values are not modified in this way and may cause error messages to be displayed when you perform certain operations (e.g., when you export query results to a file). If you see an error message, simply modify the duplicate values to be unique.

- The alias cannot exceed 27 characters in length. You will not be able to type more than 27 characters into any **Alias** cell.
- If desired, you can define translations for query aliases by placing your cursor in

the desired **Alias** cell and clicking the [icon] button to manage translations for that string. Note that you can define translations for query aliases via the design grid only. You cannot define query column aliases via the SQL code.

> **Note:** While there is a limit of 27 characters imposed on the default value for an alias, translations are not restricted to any limit.

Removing Columns from a Query

To remove a column from a query:

1. In the Meridium APM Framework, for the query from which you want to remove a column, access the query design.

2. In the design grid, select the column that you want to remove from the query.

   The selected column is highlighted in yellow.

3. Right-click the column, and click **Delete column**.

   -or-

   Press the Delete key on your keyboard.

   The column is removed from the design grid and the SQL code and will no longer appear in the results when you run the query.

Changing the Column Order

After you have added columns to a query, you can change the order of those columns by modifying their arrangement in the design gird.

> **Note:** The column order that you specify within the query definition will be saved as the default layout for the results. The default layout will be overridden if a different layout is defined via the results grid and saved with the query.

To modify the column order of a query:

1. In the Meridium APM Framework, for the query whose columns you want to reorder, access the query design.

2. In the design grid, click the column header of the column that you want to move.

3. While holding down your left mouse button, drag the column to the desired location in the grid.

4. Release the left mouse button.

   The column is moved to the desired location.

5. Continue in this way, moving other columns as needed, until you have achieved the desired column layout.

Hiding and Displaying Columns in the Results

When you create a query, all the fields that you added are selected by default to be displayed in the query results. If desired, you can hide columns from the query results. This may be necessary in cases where the field is needed in the select statement but where you do not want to display the field in the results, such as when you are adding hyperlinks that use an Entity Key or Family Key.

Note: Any column that is displayed in the query results is also included in the SQL select statement.

For example, in the following image, the **Display** check box for the **Asset ID** column is cleared.

Field	[Asset ID]	[Asset Description]	[Asset Installation Date]
Alias	Asset ID	Asset Description	Asset Installation Date
Table	[Pump]	[Pump]	[Pump]
Sort			
Sort Index			
Include	☑	☑	☑
Display	☐	☑	☑
Hyperlink			
Criteria			
Or			

The **Asset ID** column, therefore, will not appear in the output, as shown in the following image.

\<New Query\> : Select Query

Asset Description	Asset Installation Date
CUB FEED TK1260	1/1/1992
CAUSTIC CIRC EAST	1/1/1992
PRETREATER REACTOR CHARGE	1/1/1972

To hide a column in the query results:

- In the design grid of the query design, in the **Display** cell for the field that you do not want to display in the results, clear the check box.

  When you run the query, the column for which you cleared the **Display** check box will not appear.

Note that you cannot display a field unless the field is included in the SQL code. This means that if the **Include** and **Display** check boxes are *cleared* and you select the **Display** check box, the **Include** check box will be selected automatically.

Note: Columns can also be removed from the query results via the standard grid layout customization options. Even if the query definition specifies that a column should be displayed in the results, that column may be hidden when the query is run if a different layout is defined via the results grid and saved with the query.

Overview of Query Prompts

You can construct a query such that, when run, it will prompt the user to enter or select values by which the results will be filtered. For example, if you create a query that returns a list of pumps, you might add a prompt that would require the user to select a specific type of pump by which to limit the results.

You can create a prompt on a character, numeric, date, or logical field in any type of query (Select, Crosstab, Append, Update, or Delete). Like all query options, you can construct prompts by modifying the SQL code directly, but the Meridium APM Framework provides the *Prompt Builder*, which guides you step-by-step through the process of creating prompts. This documentation focuses primarily on that feature.

> **Note:** There are special considerations that you will want to take into account when creating a prompt on a date field.

When a user runs a query by opening it from the Catalog, prompts will appear in a dialog box, which disappears when the results are returned. Alternatively, you can customize URLs to specify that you want the query prompts to appear at the top of the results. Choosing this method allows the user to see the prompt values that limited the current query results.

For example, in the following image, a prompt exists on the Asset Manufacturer field to limit the query results based on the manufacturer. If this query were accessed via the Catalog or a URL that does not specify that the prompt should appear at the top of the results, a dialog box would appear, prompting the user to supply the desired value for the manufacturer:
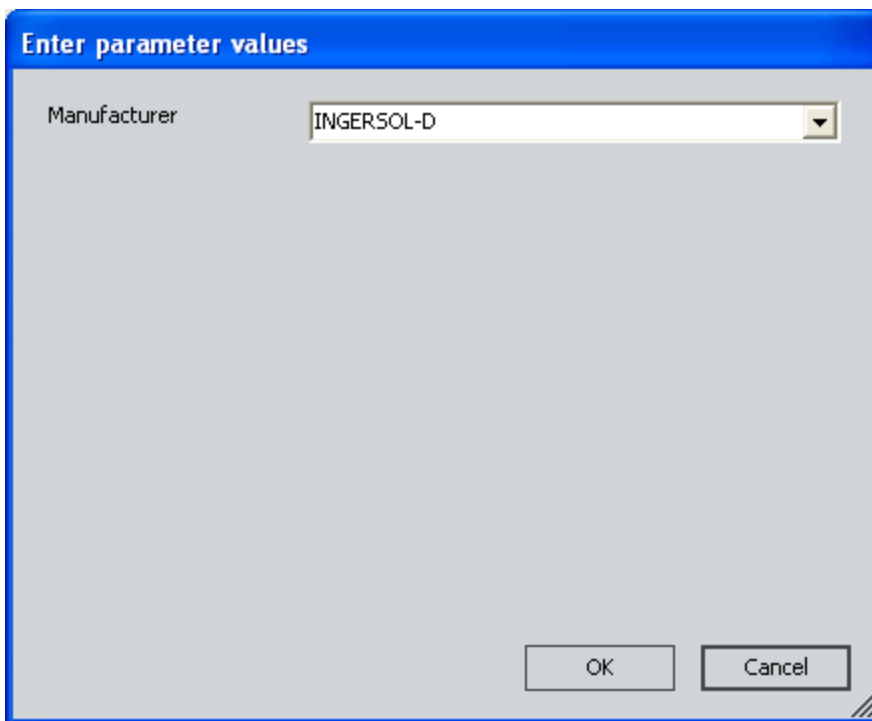
If this same query were accessed via a URL that specifies that the prompt should appear at the top of the results, the prompt would appear at the top of the results. In the following image, the query results are filtered based on the prompt value INGERSOL-D.
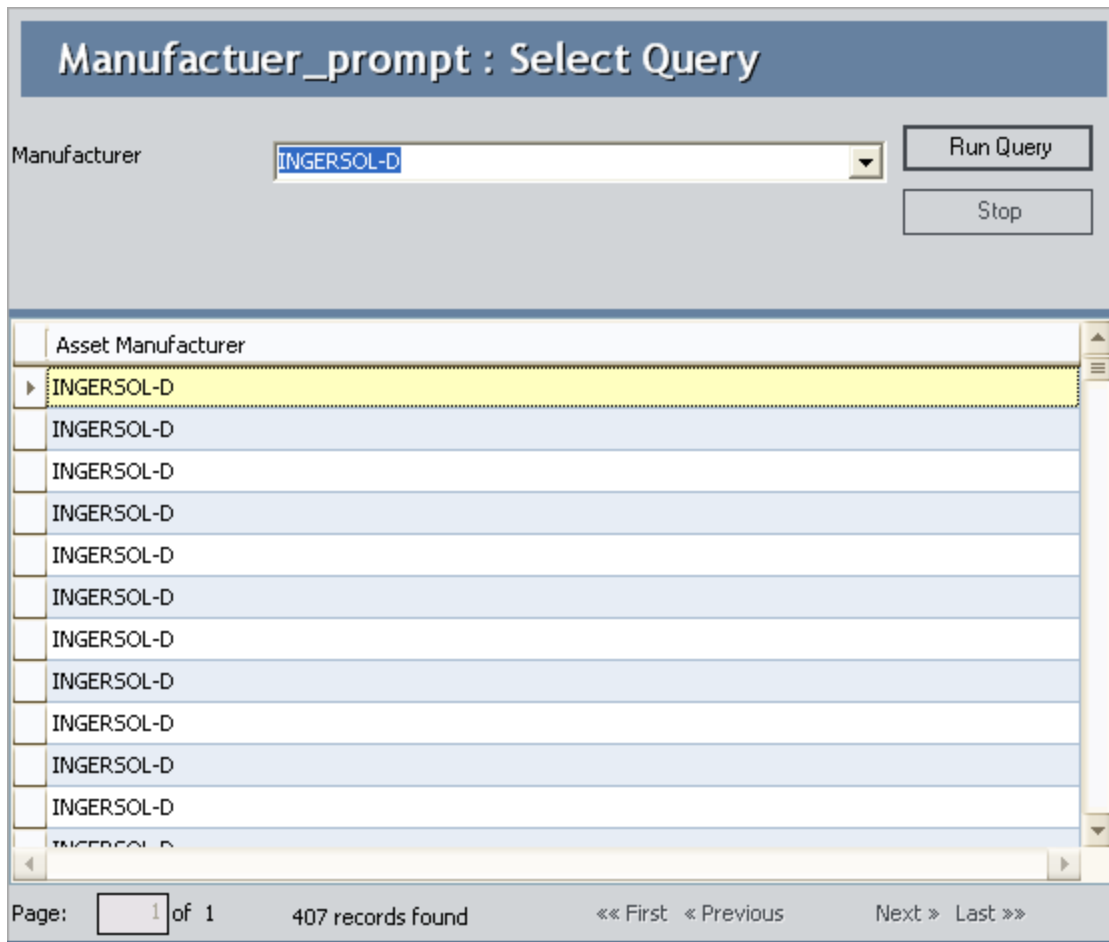
To simplify our instructions, throughout our description of building prompts, we assume that the query is run in such a way that the dialog box appears.

Accessing the Prompt Builder

The **Prompt Builder** guides you step-by-step through the process of creating new prompts.

> **IMPORTANT:** The **Prompt Builder** does not support modifying existing prompts. If you access the **Prompt Builder** for a field that already contains a prompt, the existing prompt will be *removed*. If you want to modify an existing prompt, you must do so by editing the SQL code.

To access the Prompt Builder:

1. In the Meridium APM Framework, create or open the query where you want to add a prompt.

2. In the **Criteria** cell of the field on which you want to base the prompt, click the ⋯ button.

   The **Expression Builder** appears.
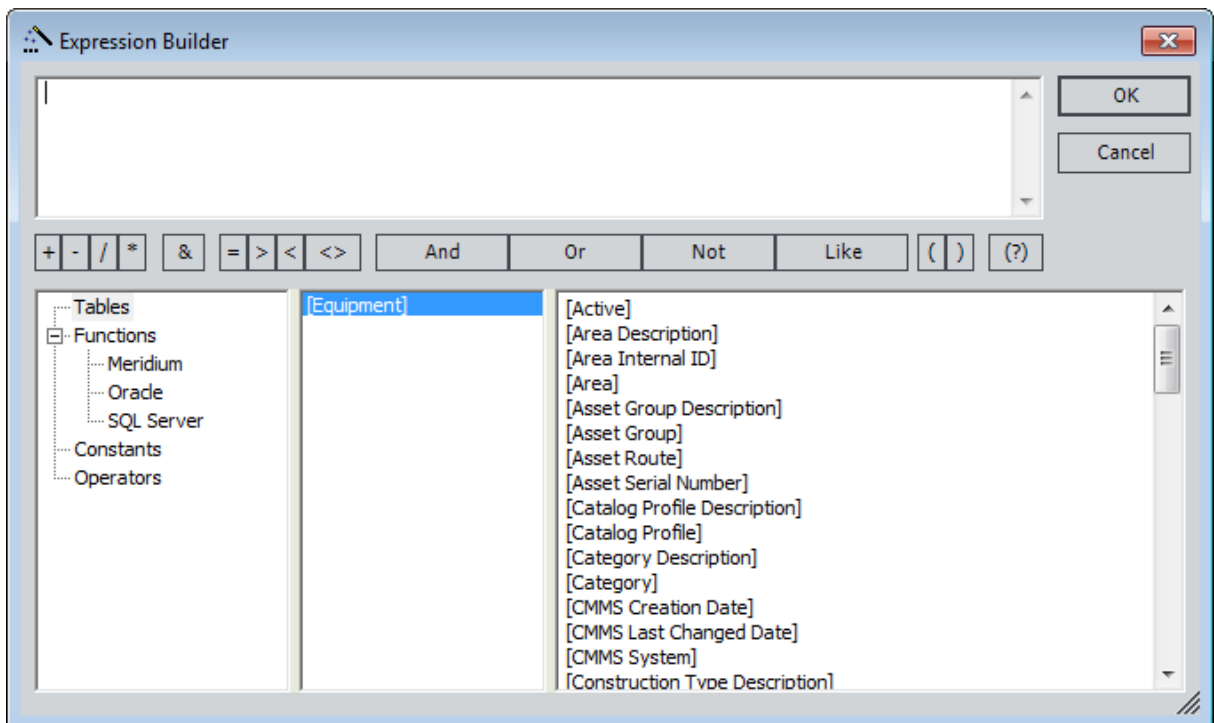
3. Click the (?) button.

   The **Prompt Builder appears**.

Prompt Settings

On the first screen of the Prompt Builder, you can define the basic settings for the prompt that you want to create.
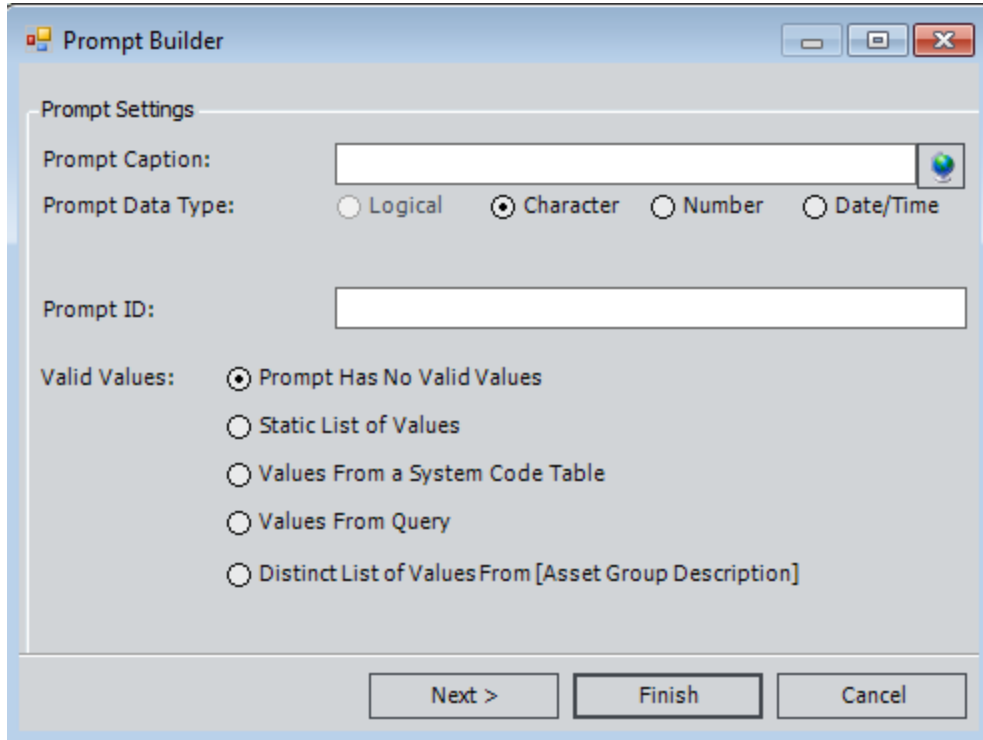
The following table provides information on the available settings. After you have defined the desired prompt settings, you can click the **Next** button to proceed through the builder and further define the details of the prompt. The content of subsequent screens in the **Prompt Builder** is dependent upon your **Valid Values** selection. These screens are documented in more detail in the topics that explain how to create specific types of prompts.

Setting	Description	Notes
Prompt Caption	A label that will indicate to the user what type of value to type or select for the prompt. The prompt caption will appear on the **Enter parameter values** dialog box to identify the prompt.	- Prompt captions are optional. - If you do *not* provide a prompt caption, the *prompt ID* will be displayed on the **Enter parameter values** dialog box, where spaces are replaced with underscores. For example, a prompt ID of *Task Type* would be displayed as *Task_Type*. - If you do not specify a prompt caption or a prompt ID, the text *Enter Parameter Value* will be displayed on the **Enter parameter values** dialog box. - If desired, you can click the button to manage translations for the prompt caption. Prompt captions can be translated via the **Prompt Builder** only and only when you are creating a prompt. You cannot translate prompt captions via the SQL code. Note that if you want to define custom translations for prompts, you must do so when you create the prompt. You cannot modify prompts via the **Prompt Builder**, and you cannot modify translations via the SQL code.

Setting	Description	Notes
Prompt Data Type	A property that identifies type of data that exists in the field *at the time the query is run*. Expressions may exist that convert the stored value to a runtime value (e.g., numeric values may be converted to character values or strings). You will want to choose a data type for the prompt that is appropriate for the *runtime* value. When you access the **Prompt Builder**, the this setting will be set by default to the data type that corresponds to the *stored* value of the field from which you accessed the Expression Builder. You may need to change the default setting. The following options are available: • **Logical:** For logical values (i.e., True and False). • **Character:** For character values. • **Number:** For numeric values. • **Date/Time:**For date values.	• When you create a prompt that allows multiple selections, you can select multiple default values for that prompt. Multiple default selections will be allowed only when the **Character** option is selected for this setting. • For [prompts on numeric fields](#), the corresponding UOM will be displayed on the **Enter parameter values** dialog box for that field if the **Number** option is selected for this setting. If you select any other option, the UOM will not be displayed.
Prompt ID	A unique, alphanumeric ID for the prompt. This value is used internally by the Meridium APM system to identify the prompt. Additionally, you will use this ID to identify the prompt in URLs.	This setting is optional, but we recommend that you specify an ID for each prompt.

Setting	Description	Notes
Valid Values	An option that defines the type of user-input value that will be required for the prompt. You can require users to specify a value manually, or you can present them with a list of valid options. Specifically, you have the following options for determining user-input values: • No Valid Values: The prompt will not offer a list of values. Rather, the prompt will display a text box, into which users can type the value by which they want to limit the results. To limit the query in this way, users will need to be familiar with the exact values that exist in the limiting field. • Static List of Values: The prompt will display a static list of values that you define specifically for the prompt. • Values from a System Code Table: The prompt will display a list of values pulled from a selected table of System Codes. • Values from Query: The prompt will display a list of values taken from the query results of a query that you select. • Distinct List of Values From [X]: The prompt will display a list of values pulled from field X of all records in a given family.	When the **Logical** option is selected for the **Prompt Data Type** selection, the Valid Values options will be *disabled*, and a list of valid values will be created automatically, providing the options *True*, *False*, and *All* (i.e., both True and False) when the query is run.
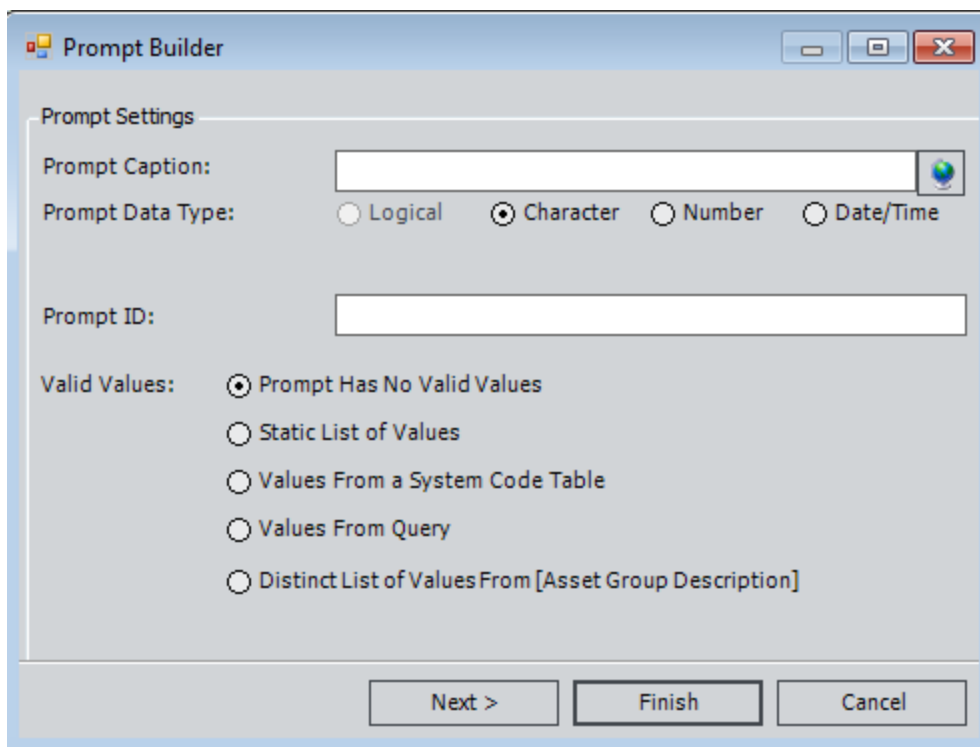
Prompt with No List of Valid Values

When you build a prompt that presents no list of values, the prompt will display a text box, into which the user can type a value to limit the query results. For example, you could provide a prompt that presents no values for the Asset Type in a query that returns pumps. When presented with the prompt, the user would have to type an Asset Type in the text box. To limit the query results in this way, the user will need to be familiar with the exact values that exist in the limiting field.

To build a prompt that does not present a list of values:

1. In the Meridium APM Framework, in the query where you want to add the prompt, access the **Prompt Builder** for the desired field.

    The **Prompt Builder** appears.

2. Define the prompt settings, as desired.

3. In the list of **Valid Values** options, select **Prompt Has No Valid Values**.

4. Click the **Next** button.

    The **Configure Default Prompt Value** screen appears.
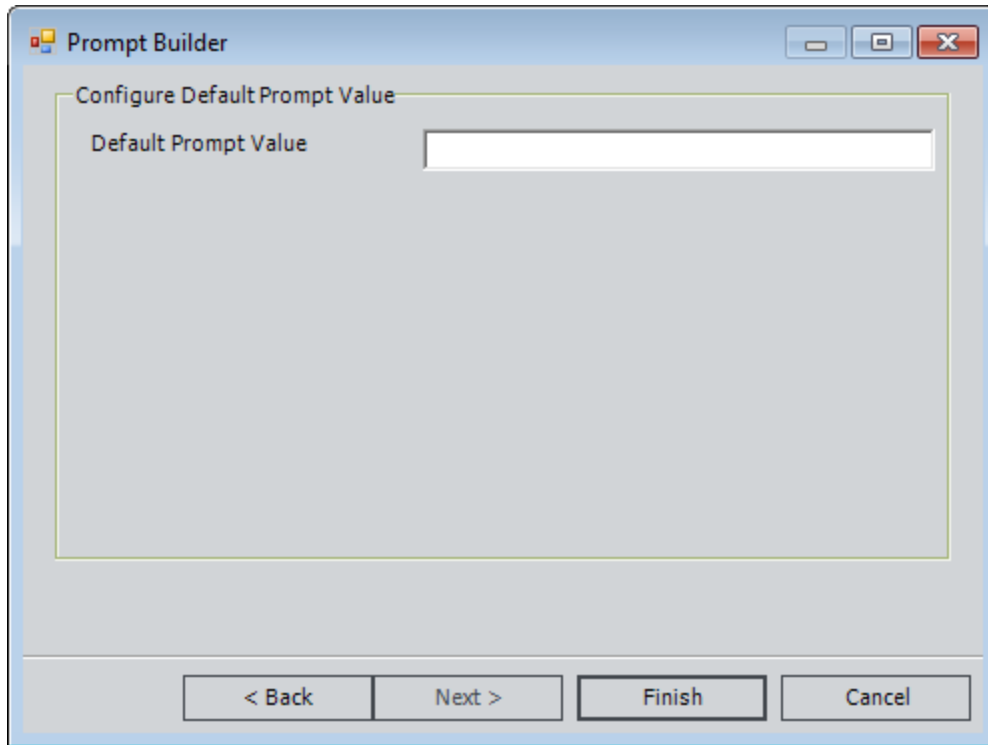
5. To provide a default value for the prompt, type the value in the **Default Prompt Value** text box.

> **Note:** If you do not want to provide a default value for the prompt, you can click the **Finish** button without typing a value in the text box.

6. Click the **Finish** button.

   The prompt expression appears in the **Expression Builder**.

7. Click **OK**.

   The **Expression Builder** closes, and the design grid displays the syntax for the prompt.

   When a user runs the query, the prompt will appear. If you specified a default prompt value, that value will appear in the text box. The user can type a different value if desired. If the user types a value that does not exist in the limiting field, the query will not return any results. For example, if two Asset Types exist in the results, such as Rotating Pump and Centrifugal Pump, and the user types **Reciprocating Pump**, the query results will be empty.
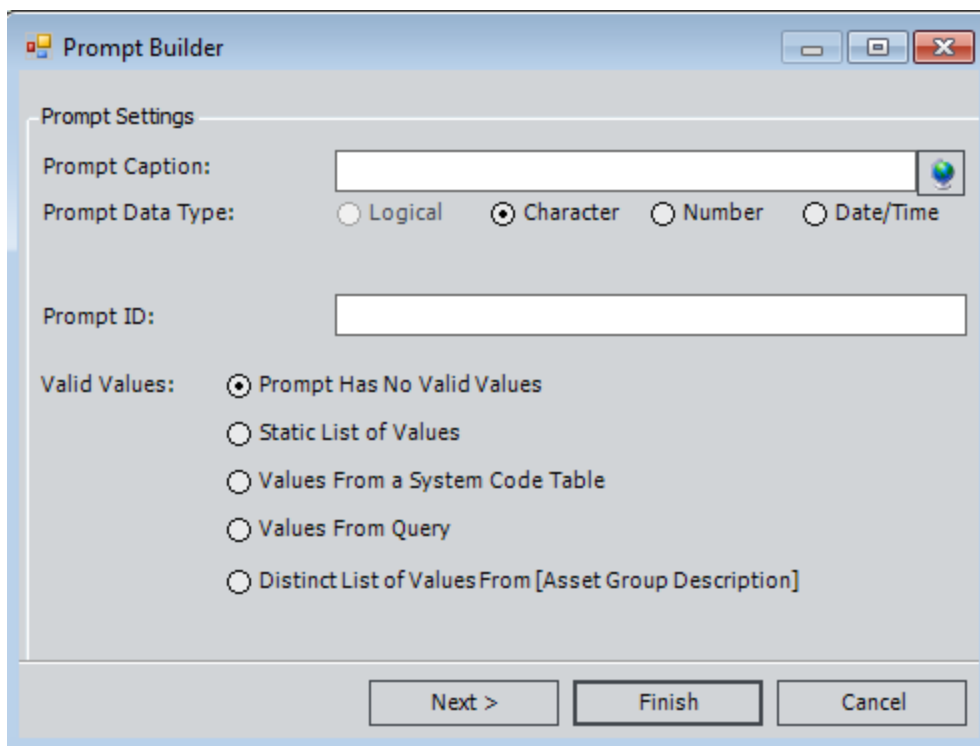
Prompt with a Static List of User-defined Values

When you build a prompt that presents a static list of pre-defined values, the prompt will display a static list of values that you define specifically for the prompt. For example, you could build a prompt that presents three pre-defined values for the Asset Type in a query that returns Pumps, such as Rotating Pump, Reciprocating Pump, and Centrifugal Pump.

To build a prompt that presents a static list of pre-defined values:

1. In the Meridium APM Framework, in the query where you want to add the prompt, access the **Prompt Builder**.

   The **Prompt Builder** appears.

2. Define the prompt settings, as desired.

3. In the list of **Valid Values** options, select **Static List of Values**.

4. Click the **Next** button.

   The **Prompt Builder** displays the next step.
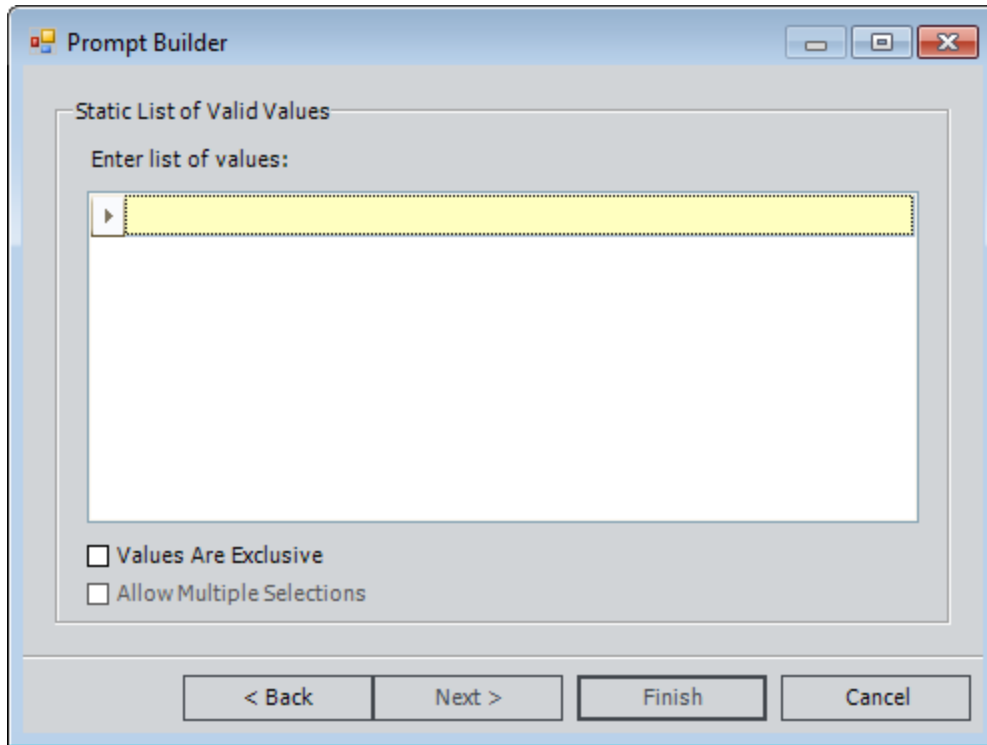
5. In the highlighted row, type the first value that you want to appear in the list. When you type the first value, press the Tab key to display a new blank row in the list. Continue adding values as needed.

> **Note:** If you are defining a list of values for numeric field and you type a non-numeric character in the list, an error message will appear when you navigate away from the cell. You will not be able to proceed or close the **Prompt Builder** until you replace the non-numeric character with a numeric one.

6. Select the **Values Are Exclusive** check box if you want users to be able to choose only from the list of defined values in the prompt dialog box. If you do not select this check box, users will be able to type an alternate value for the prompt.

7. If you selected the **Values Are Exclusive** check box, select the **Allow Multiple Selections** check box if you want users to be able to choose multiple values by which to filter the query results.

> **Note:** The **Allow Multiple Selections** check box is enabled only if you select the **Values Are Exclusive** check box.

8. Click the **Next** button.

The **Prompt Builder** displays the next step.
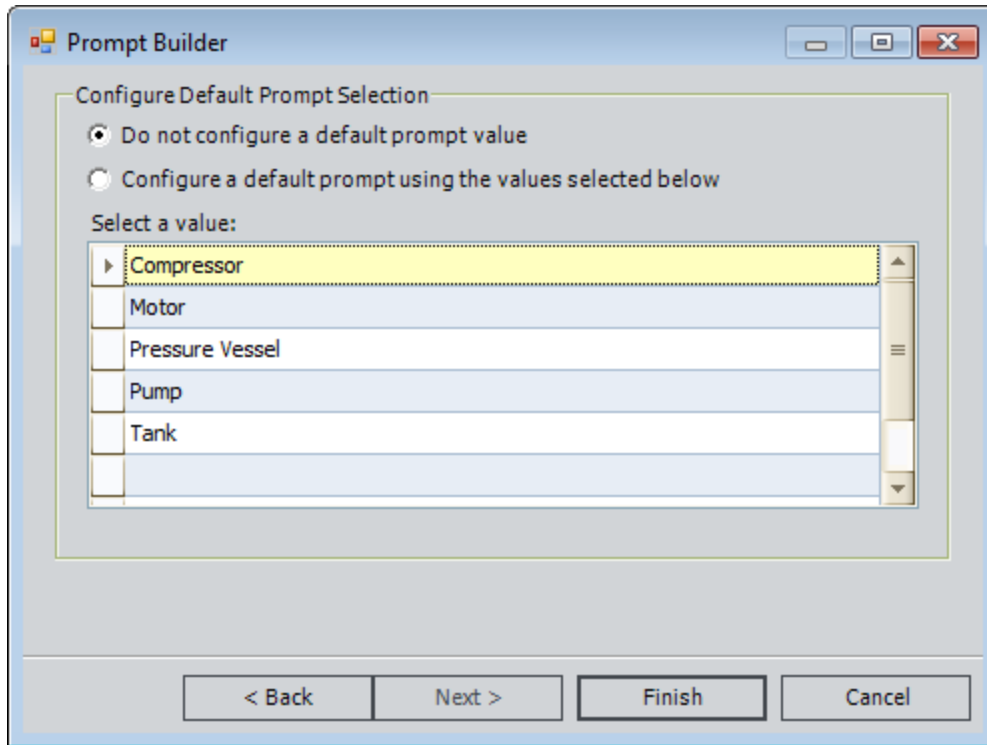
9. If you want the first value in the list in the query prompt to be selected by default, accept the default selection, **Do not configure a default prompt value**.

   -or-

   If you want to choose a value that will be selected by default in the query prompt, in the grid, select the row containing the value that you want to be selected by default. The **Configure a default prompt using the value selected below** option will be selected automatically.

   > **Note:** If the Prompt Data Type is set to *Text* and you selected the **Allow Multiple Selections** check box on the previous screen, you can select multiple default values for the prompt using the Ctrl or Shift key.

10. Click the **Finish** button.

    The prompt expression appears in the **Expression Builder**.

11. Click **OK**.

    The design grid displays the syntax for the prompt.

    When a user runs the query, the prompt will display a list of the pre-defined values from which to select.
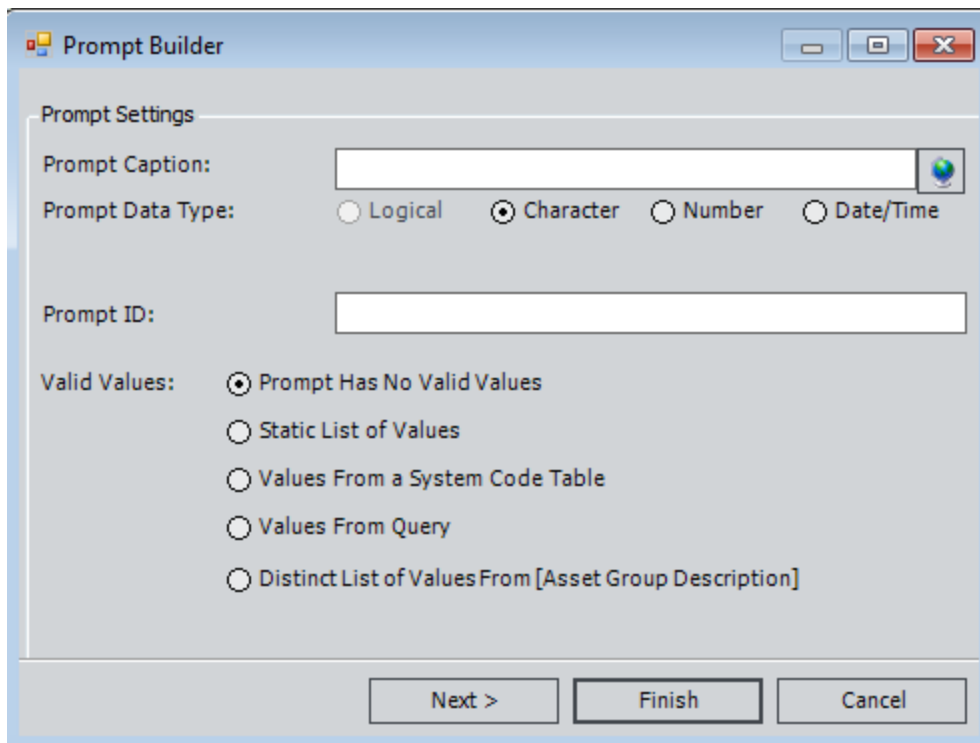
Prompt with a List of System Codes

When you build a prompt that displays a list of System Codes, the prompt will display a list of values pulled from a selected Table of System Codes. For example, you could build a prompt for the Critical Asset? field that presents values Yes and No from the system code table M227 in a query that returns Pumps.

To build a prompt that displays a list of System Codes:

1. In the Meridium APM Framework, in the query where you want to add the prompt, access the **Prompt Builder**.

   The **Prompt Builder** appears.

2. Define the prompt settings, as desired.

3. In the list of **Valid Values** options, select **Values from a System Code Table**.

4. Click the **Next** button.

   The **Prompt Builder** displays the next step.
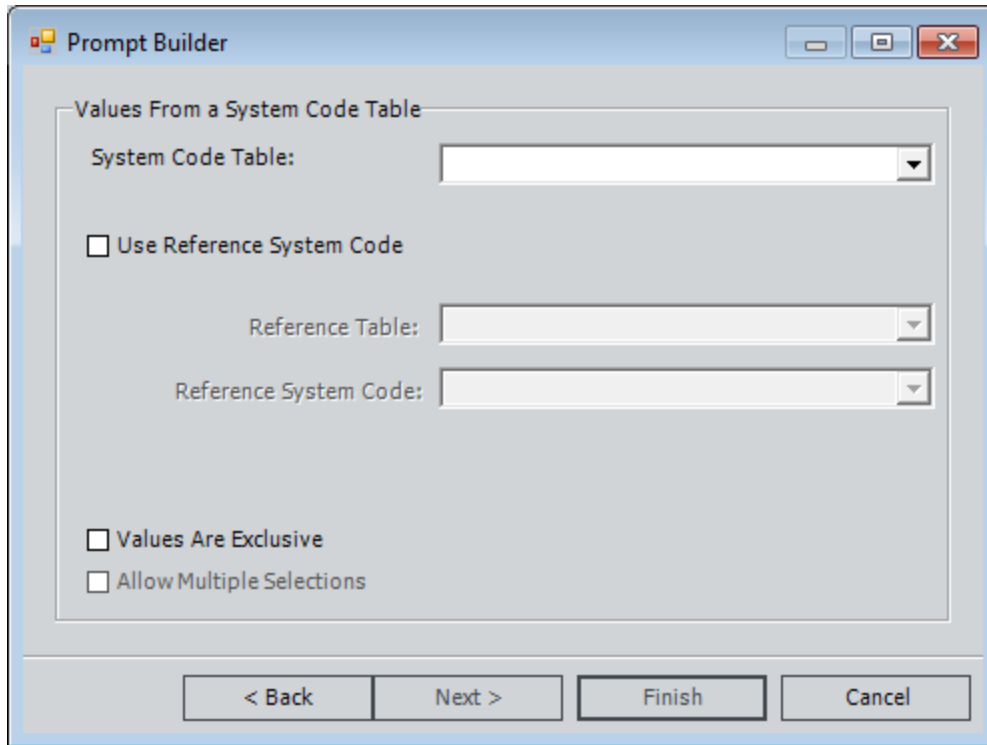
5. In the **System Code Table** list, select the System Code Table whose values you want to display in the prompt.

6. If desired, select the **Use Reference System Code** check box to specify a referenced System Code Table (i.e., a System Code Table that is referenced within the original table in the Configuration Manager).

7. Select the **Values Are Exclusive** check box if you want users to be able to choose only from the list of presented values in the prompt dialog box. If you do not select this check box, users will be able to type an alternate value for the prompt.

8. If you selected the **Values Are Exclusive** check box, select the **Allow Multiple Selections** check box if you want users to be able to choose multiple values by which to filter the query results.

Note: The **Allow Multiple Selections** check box is enabled only if you select the **Values Are Exclusive** check box.

9. Click the **Next** button.

The **Configure Default Prompt Selection** screen appears.
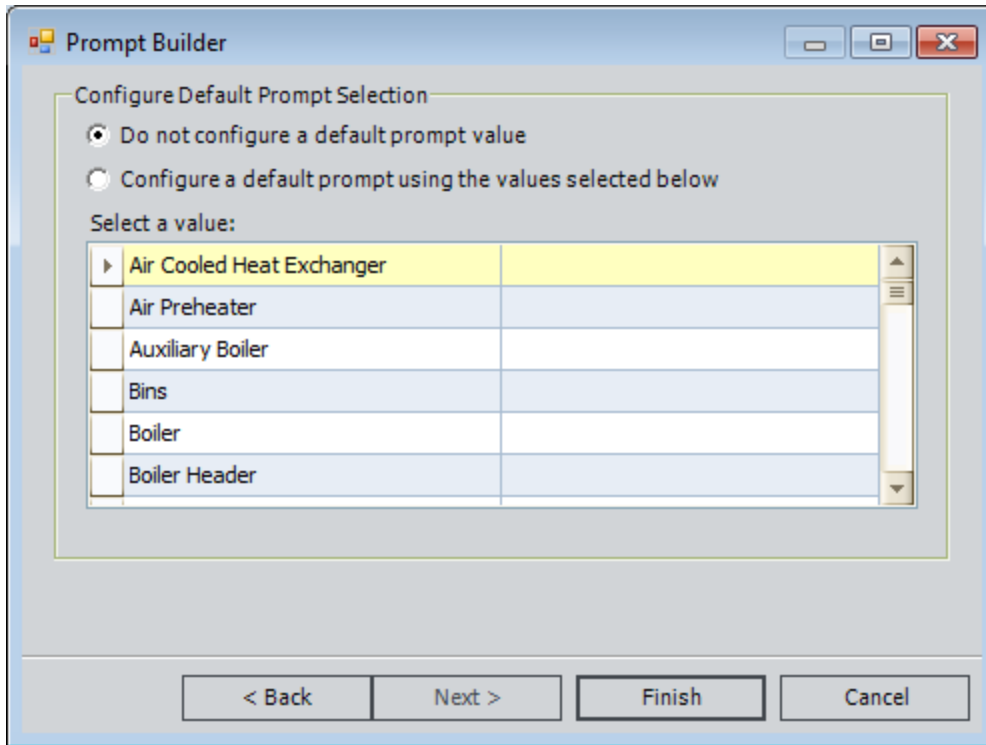
10. If you want the first value in the list in the query prompt to be selected by default, accept the default selection, **Do not configure a default prompt value**.

    -or-

    If you want to choose a value that will be selected by default in the query prompt, in the grid, select the row containing the value that you want to be selected by default. The **Configure a default prompt using the value selected below** option will be selected automatically.

    > **Note:** If the Prompt Data Type is set to *Character* and you selected the **Allow Multiple Selections** check box on the previous screen, you can select multiple default values for the prompt using the Ctrl or Shift key.

11. Click the **Finish** button.

    The prompt expression appears in the **Expression Builder**.

12. Click **OK**.

    The design grid displays the syntax for the prompt.

    When users run the query, the prompt will contain a list of values as defined in the associated System Code Table.

Prompt with a List of Query Results

When you create a prompt that presents a list of results from another query, the prompt selection dialog box will display a drop-down list of values retrieved by running that query.

For example, suppose the pieces of equipment in your company are manufactured by five different companies. Each of those manufacturers has a Manufacturer record, which stores the Manufacturer ID, Name, and other identifying information. Also, suppose each Equipment record in your database also stores the manufacturer ID in the Manufacturer field. The Equipment records are not linked to the Manufacturer records.

Suppose you want to create a query that returns a list of equipment manufactured by a certain company, but you do not know the exact manufacturer ID for that company. Because the Equipment family contains only the Manufacturer field to identify the manufacturer ID, querying just the Equipment family would not allow you to add a prompt on a more commonly known value, such as the manufacturer name.

In this case, you could create a query on the Equipment family, add the Manufacturer field, and create a prompt on that field that retrieves values from a separate query. This separate query could query the Manufacturer family and return the Manufacturer ID and the Manufacturer Name. As long as the Manufacturer ID is the first column and the Manufacturer Name is the second column in the Manufacturer query, when you add a prompt to the Manufacturer ID field in the equipment query, it will display values in the Manufacturer Name field in the Manufacturer records.

Note that you can use only a Select query for generating a list of prompt values. Note also that:

- If the selected query contains only ONE column, the values returned in that column will be displayed in the list of available prompt values. For example, if the query contains the Equipment ID column, the prompt will display a list of Equipment IDs.

- If the selected query contains *more than one* column, the list of prompt values will contain a concatenated list of values from the *second* column and each subsequent column. For example, if the query results look like this table:

Equipment ID	Manufacturer	Description	Status
123	Alco	Tank	Active
456	Whitlock	Pump	Inactive
789	Delta	Pressure Vessel	Active

...the list of prompt values would look like this:

- Alco Tank Active
- Whitlock Pump Inactive

- Delta Pressure Vessel Active
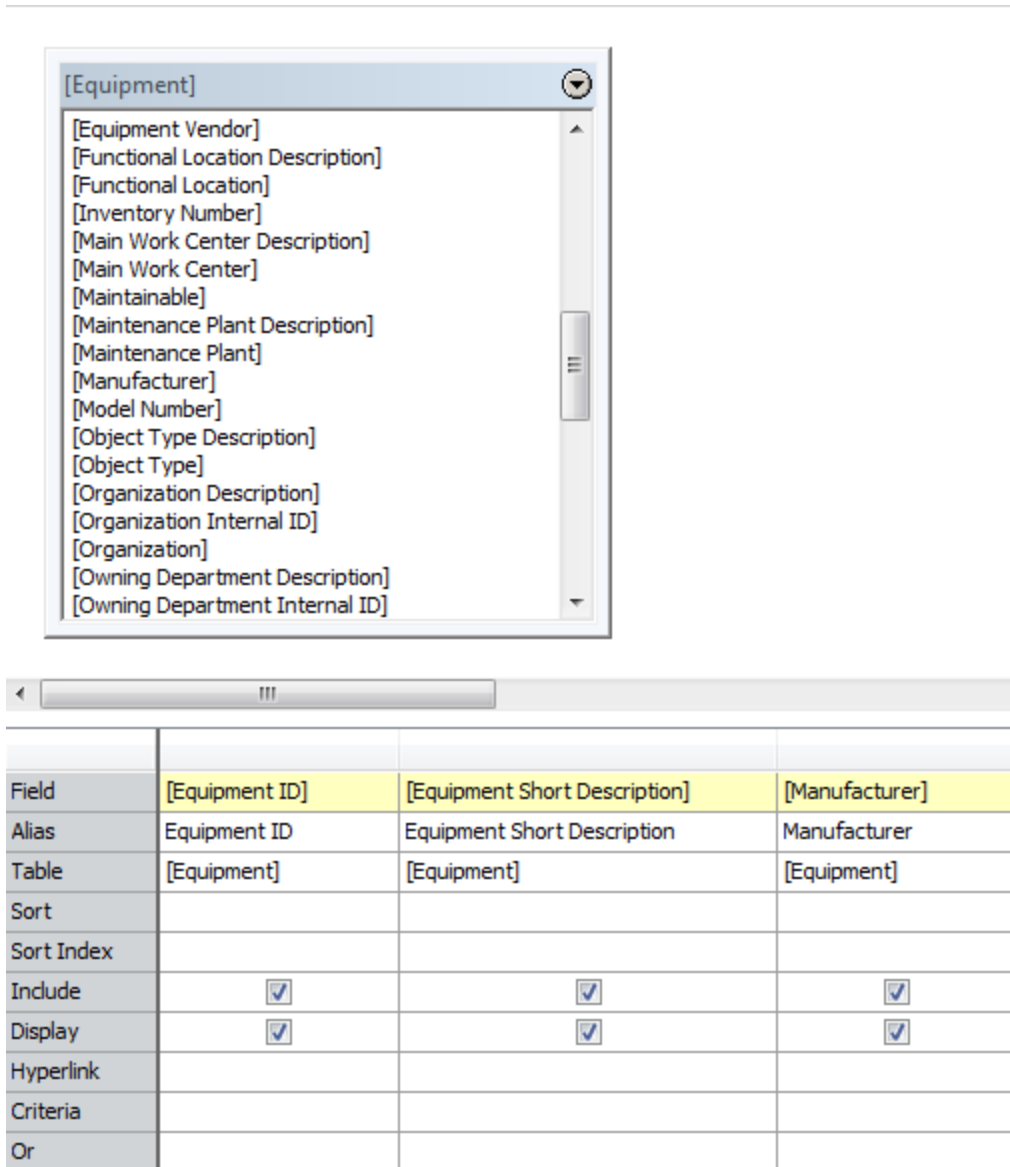
    Note that the values from the first column, Equipment ID, are not displayed as part of the prompt values.

> Œ Hint: Select the **Unique Values Only** option in the prompt query to keep the prompt list from containing duplicate values.

Because the concept of building a query prompt based on another query is best understood using a specific example, the following instructions explain how to build a prompt based on a query using the preceding equipment manufacturer example.

To build a prompt that displays a list of query results:

1. In the Meridium APM Framework, open the query to which you want to add a prompt. Using the equipment manufacturer example, this query would be configured as shown in the following image.

[Equipment]

[Equipment Vendor]
[Functional Location Description]
[Functional Location]
[Inventory Number]
[Main Work Center Description]
[Main Work Center]
[Maintainable]
[Maintenance Plant Description]
[Maintenance Plant]
[Manufacturer]
[Model Number]
[Object Type Description]
[Object Type]
[Organization Description]
[Organization Internal ID]
[Organization]
[Owning Department Description]
[Owning Department Internal ID]
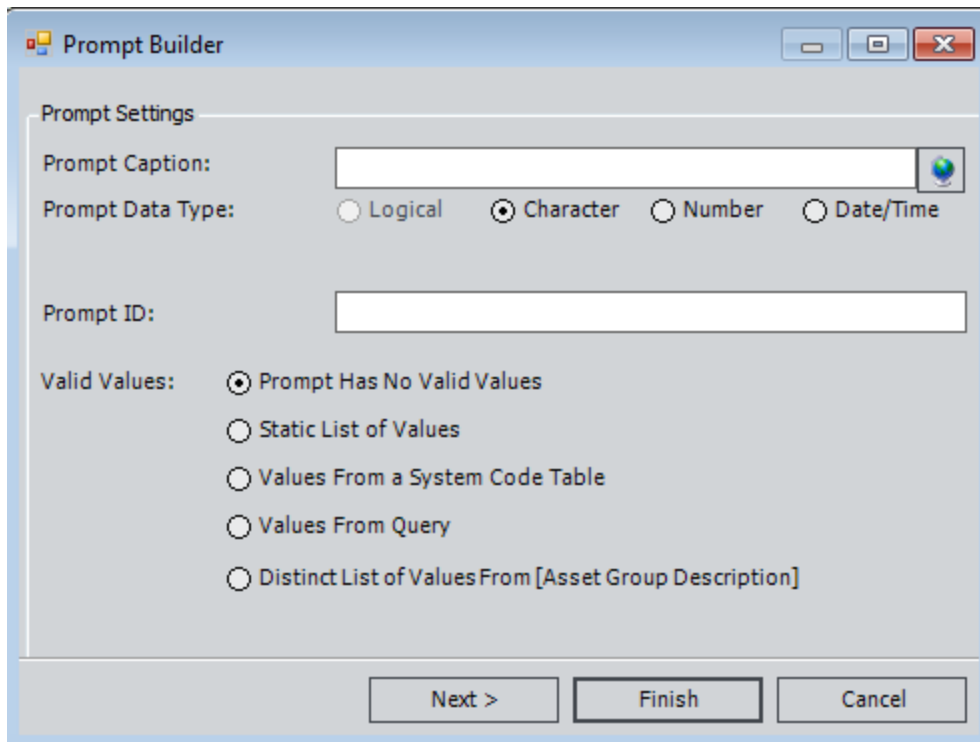
	[Equipment ID]	[Equipment Short Description]	[Manufacturer]
Field	[Equipment ID]	[Equipment Short Description]	[Manufacturer]
Alias	Equipment ID	Equipment Short Description	Manufacturer
Table	[Equipment]	[Equipment]	[Equipment]
Sort			
Sort Index			
Include	✓	✓	✓
Display	✓	✓	✓
Hyperlink			
Criteria			
Or			

Note that the query includes the Equipment family, and the **Equipment ID**, **Equipment Short Description**, and **Manufacturer** fields are included in the design grid. You would access the **Prompt Builder** for the **Manufacturer** field.

2. [Access the **Prompt Builder**](#).

   The **Prompt Builder** appears.

3. Define the prompt settings, as desired.

4. In the list of **Valid Values** options, select **Values from Query**.

5. Click the **Next** button.

   The **Values From Query** screen appears.
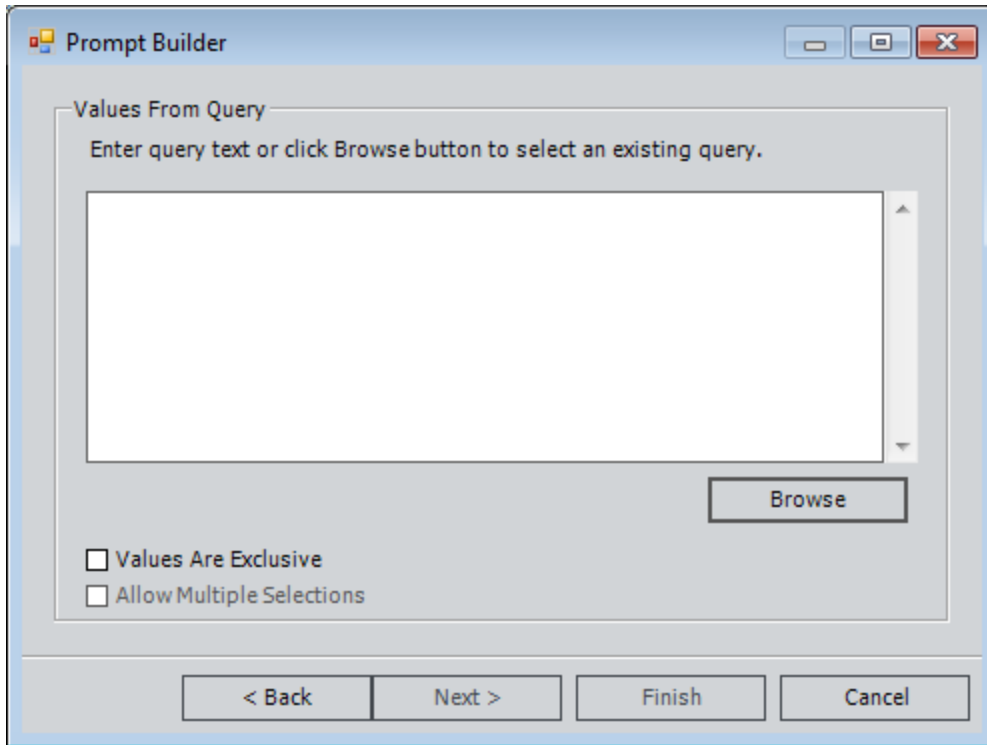
6. Click the **Browse** button.

   The **Choose a stored query** dialog box appears.

7. Navigate to the query whose results you want to offer as values in the prompt, and click the **Open** button.

   > Note: Be sure to choose a Select query. The Meridium APM system will allow you choose an Update, Delete, or Append query in the **Prompt Builder**, but when you try to run the query, an error will appear.

   In the equipment manufacturer example, the query would be configured as shown in the following image.

Field	[Manufacturer ID]	▼ ⋯	[Manufacturer Name]
Alias	Manufacturer ID		Manufacturer Name
Table	[Manufacturer]		[Manufacturer]
Sort			
Sort Index			
Include	☑		☑
Display	☑		☑
Hyperlink			
Criteria			
Or			
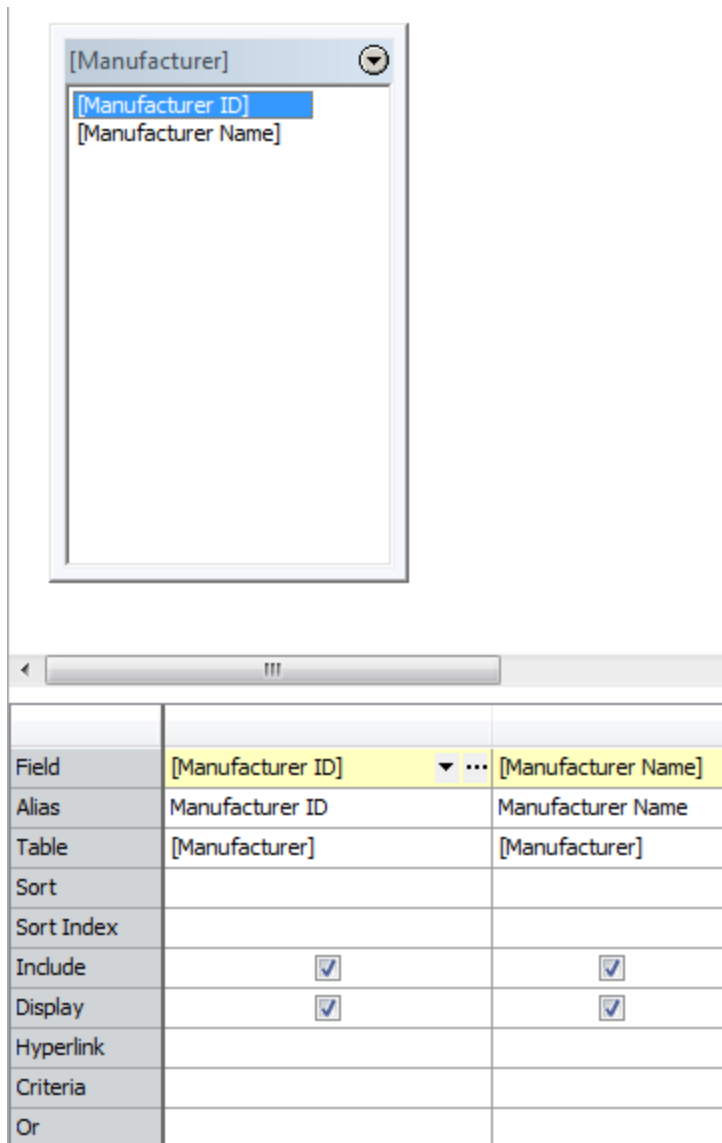
Note that the query includes the Manufacturer family, and the **Manufacturer ID** and **Manufacturer Name** fields are included in the design grid. The prompt in the equipment query is configured on the **Manufacturer** field, and it will, by default, return a list of values from the *first column* of the Manufacturer query. This means that the **Manufacturer ID** field *must* exist in the first column so the correct values will be passed to the query prompt.

8. Select the **Values Are Exclusive** check box if you want users to be able to choose only from the list of query results. If you do not select this check box, users will be able to type in an alternate value in the prompt.

9. If you selected the **Values Are Exclusive** check box, select the **Allow Multiple Selections** check box if you want users to be able to choose multiple values by which to filter the query results.

> **Note:** The **Allow Multiple Selections** check box is enabled only if you select the **Values Are Exclusive** check box.

10. Click the **Next** button.

    The **Configure Default Prompt Selection** screen appears.

11. If you want the first value in the list in the query prompt to be selected by default, accept the default selection, **Do not configure a default prompt value**.

    -or-

    If you want to choose a value that will be selected by default in the query prompt, in the grid, select the row containing the value that you want to be selected by default. The **Configure a default prompt using the value selected below** option will be selected automatically.

> **Note:** If the [Prompt Data Type](#) is set to *Text* and you selected the **Allow Multiple Selections** check box on the previous screen, you can select multiple default values for the prompt using the Ctrl or Shift key.

12. Click the **Finish** button.

    The prompt expression appears in the **Expression Builder**.

13. Click **OK**.

    The **Expression Builder** closes. Using the example, when a user runs the query, the prompt will contain a list of values from the *Manufacturer Name* field in all Manufacturer records. The query results will display all Equipment records where the value in the *Manufacturer* field value is equal to the Manufacturer ID that corresponds to the Manufacturer Name you selected in the prompt.

Prompt with a List of Values from Records
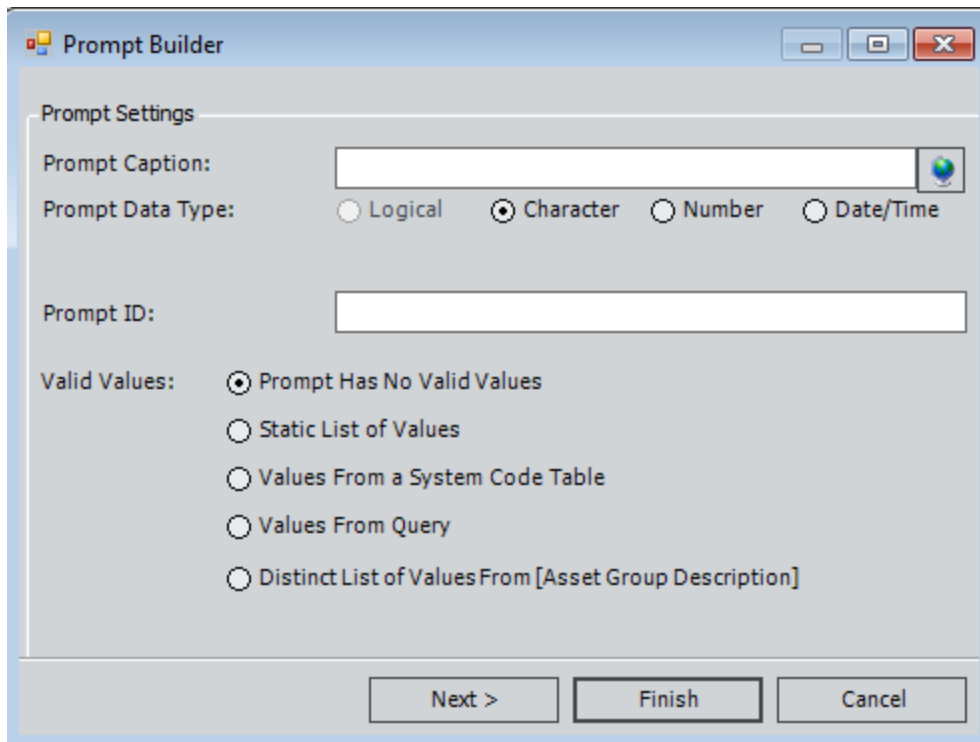
When you build a prompt that presents a list of fields from a table, the prompt will display a list of values pulled from the specified field of all records in a given family. For example, you could build a prompt that contains values pulled from the Taxonomy Type field of all records in the Equipment family.

You can create this type of prompt by selecting the **Distinct List of Values from [X]** option in the **Prompt Builder**. When you select the **Logical** option for the **Prompt Data Type** setting, the **Distinct List of Values from [X]** option will be selected by default. This will cause the prompt to display a list that contains two values: *True* and *False*. These are the only values that will ever exist in logical fields.

To build a prompt that displays a list of fields from a table:

1. In the Meridium APM Framework, in the query where you want to add the prompt, access the **Prompt Builder**.

   The **Prompt Builder** appears.

2. Define the prompt settings, as desired.

3. In the list of **Valid Values** options, select **Distinct List of Value from [X]**, where **X** is the name of the field for which you are defining the prompt criteria.

4. Click the **Next** button.

   The **Values From Query** screen appears.
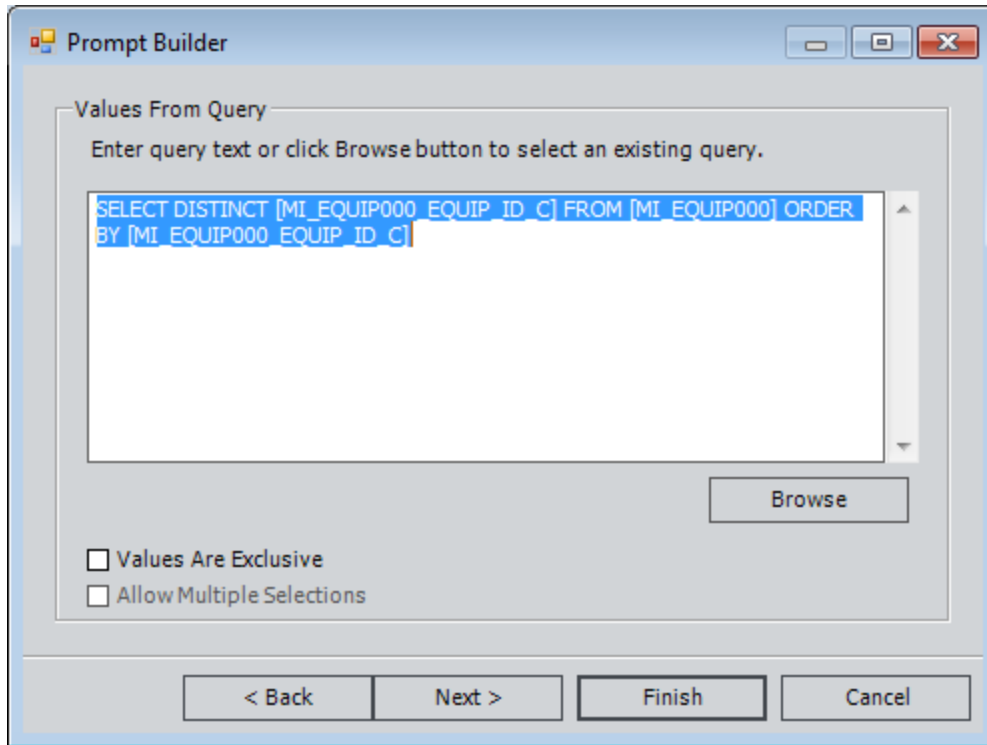
5. Select the **Values Are Exclusive** check box if you want users to be able to choose only from the list of query results in the prompt dialog box. If you do not select this check box, users will be able to type an alternate value in the prompt.

6. If you selected the **Values Are Exclusive** check box, select the **Allow Multiple Selections** check box if you want users to be able to choose multiple values by which to filter the query results.

   > Note: The **Allow Multiple Selections** check box is enabled only if you select the **Values Are Exclusive** check box.

7. Click the **Next** button.

   The **Configure Default Prompt Selection** screen appears.
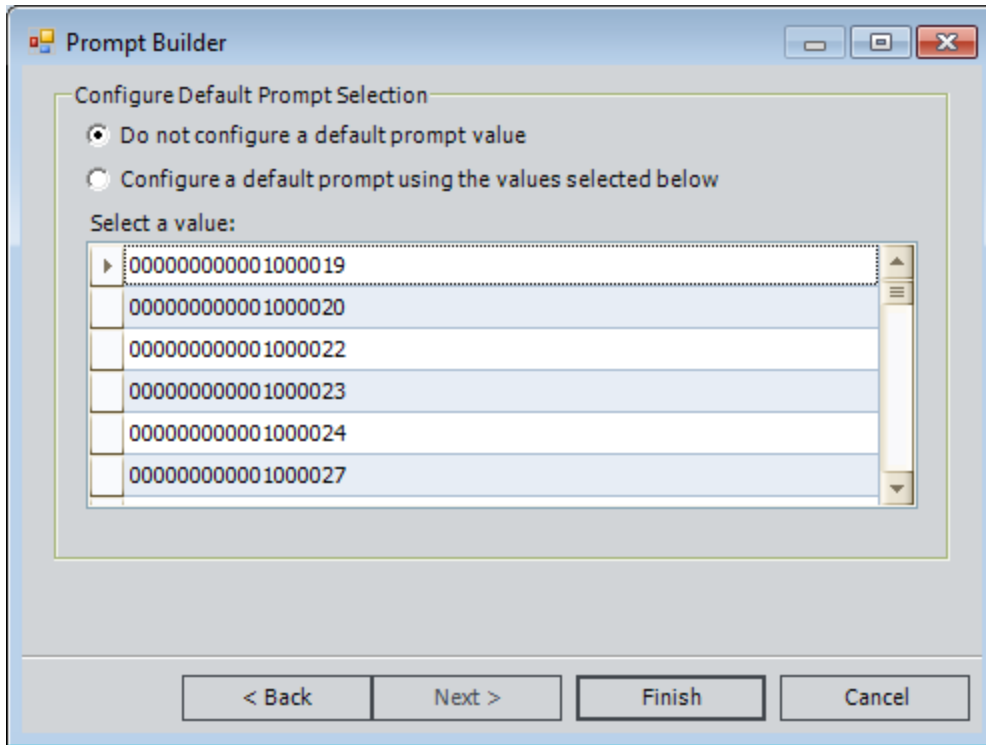
8. If you want the first value in the list in the query prompt to be selected by default, accept the default selection, **Do not configure a default prompt value**.

   -or-

   If you want to choose a value that will be selected by default in the query prompt, in the grid, select the row containing the value that you want to be selected by default. The **Configure a default prompt using the value selected below** option will be selected automatically.

   > **Note:** If the Prompt Data Type is set to *Text* and you selected the **Allow Multiple Selections** check box on the previous screen, you can select multiple default values for the prompt using the Ctrl or Shift key.

9. Click the **Finish** button.

   The prompt expression appears in the **Expression Builder**.

10. Click **OK**.

    The **Expression Builder** closes, and the design grid displays the prompt syntax.
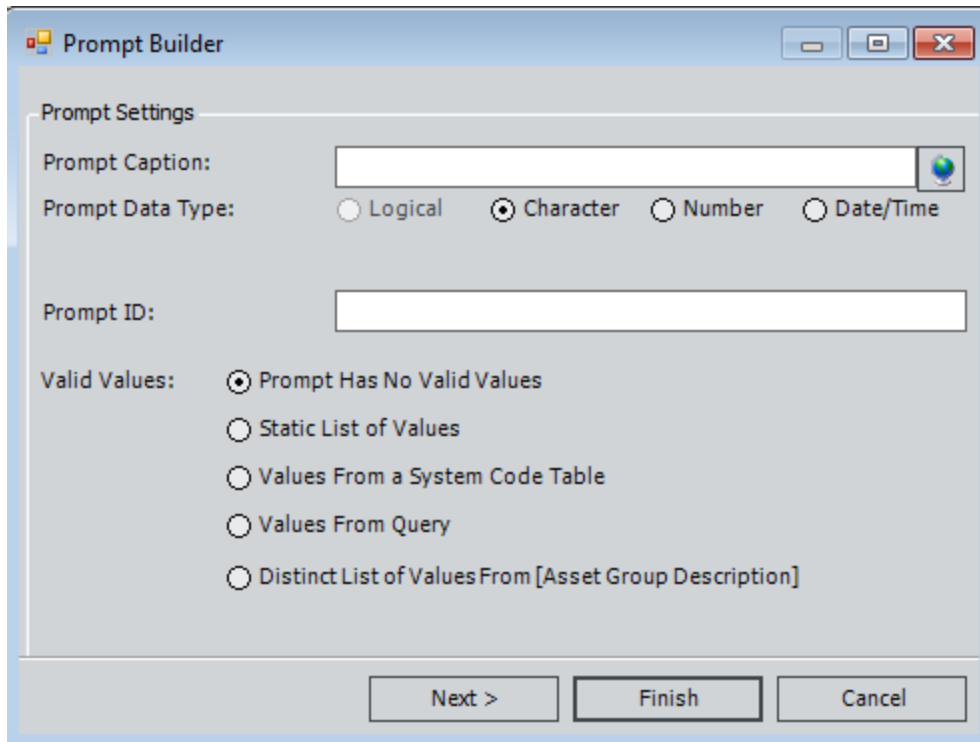
    When the user runs the query, the prompt will display a list of values pulled from the selected field for all records in the selected family.

Prompt on a Logical Field

To build a prompt for a logical field:

1. In the Meridium APM Framework, in the query where you want to add the prompt, access the Prompt Builder.

   The **Prompt Builder** appears.

2. Type a value in the **Prompt Caption** text box.

3. For the **Prompt Data Type** option, accept the default selection: **Logical**.

4. If desired, type a value in the **Prompt ID** text box.

5. Click the **Next** button.

   The **Configure Default Prompt Value** screen appears.
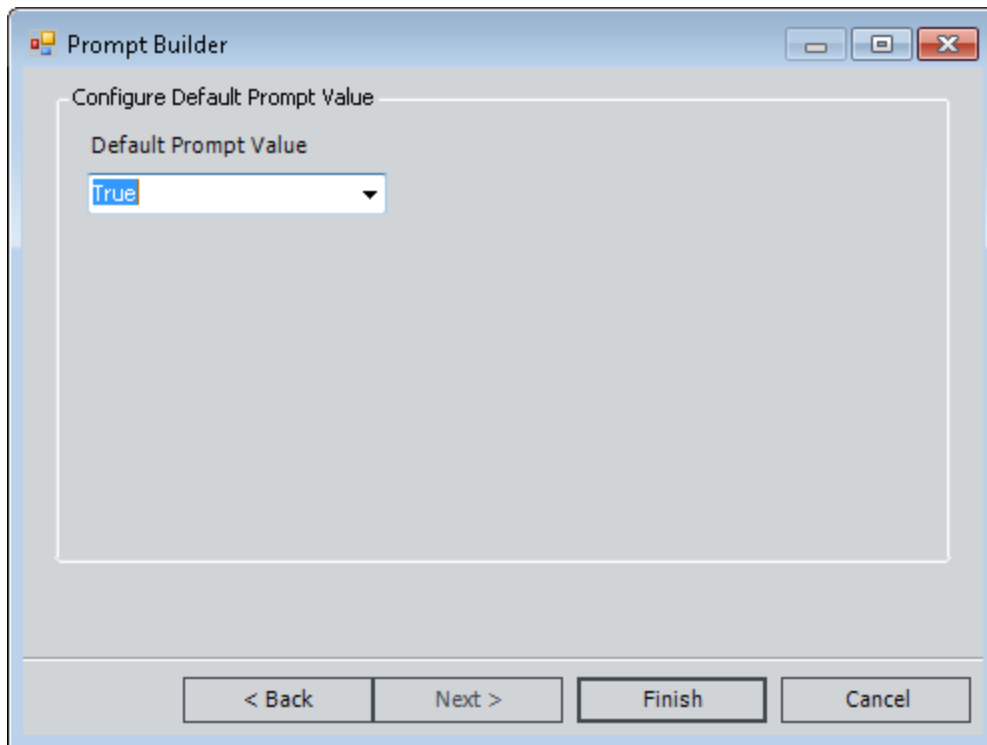
6. In the **Default Prompt Value** list, select the value that you want to be selected by default. You can choose *True*, *False*, or *All* (returns both True and False).

7. Click the **Finish** button.

   The prompt expression appears in the **Expression Builder.**

8. Click **OK.**

   The **Expression Builder** closes, and the design grid displays the prompt syntax.

   When the user runs the query, the prompt will display a list of values pulled from the selected field for all records in the selected family.

Filtering a Prompt's Values Based on Previous Prompt Selections

You can set up multiple prompts where the value in one prompt filters the values available in another prompt, which would further refine the records that are returned. For example, you could configure a State prompt and a City prompt where the only cities available in the City prompt are those that are associated with the selected state. So, if you select *Virginia* in a State prompt, a dependent City prompt would then display only the cities that exist in records where the State field contains the value Virginia.

The following instructions provide details on filtering prompt values by continuing with the State and City example we introduced. Note that these instructions assume that you have already created a Select query for a family that contains a City and State field and that you have added those fields to your query.

Note that when creating the dependent prompt, you must specify that the prompt use either the **Values from a Query** option or the **Distinct List of Values From <Field>** option.

To create a prompt whose values are filtered based on previous prompt selections:

1. In the Meridium APM Framework, in the desired query, create the first prompt whose values will filter the values in the second prompt. The first prompt should exist in the design grid in a column that occurs to the *left* of the column containing the field you will use for the *second* prompt.

   For example, if you wanted the value specified for the State prompt to filter the values available for the City prompt, you would create the prompt on the State field first, and the State field would exist in the design grid to the left of the City field. If you do not order the fields this way, the values in the State prompt will still filter the values available in the City prompt, but the City prompt will appear *first* when you run the query, which does not indicate a natural workflow.

   For the first prompt, you can specify a static list of values, you can specify that the values come from a distinct list of values or a query, or you can specify that no values will be presented in the prompt, in which case the user must type the exact value stored in the database in order to populate the dependent prompt with the appropriate values.
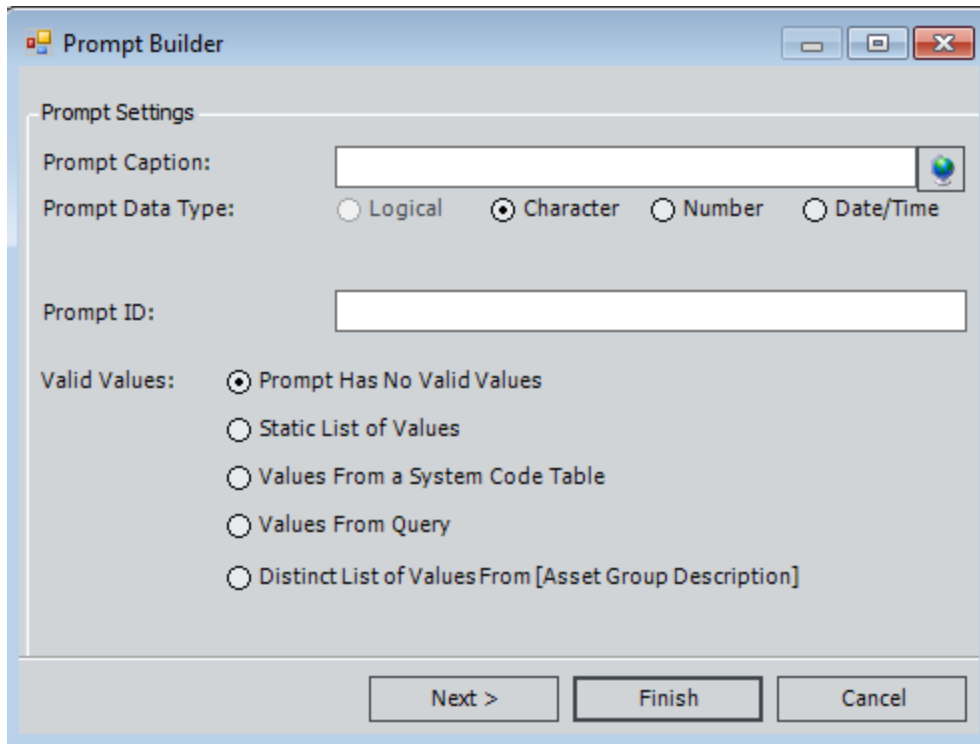
   For the purposes of these instructions, we assume that you type **State** as the Prompt ID for the State prompt, which will be used later in the SQL code.

2. In the **Criteria** cell on the field that you want to use as the *second* prompt, click the ⋯ button. For example, if you wanted the values for the City prompt to be limited based upon the value selected in the State prompt, you would click the ⋯ button in the **Criteria** cell on the **City** field to create the City prompt.

   The **Expression Builder** appears.

3. In the **Expression Builder**, click the **user prompt** button.

The **Prompt Builder** appears.

4. On the first screen of the **Prompt Builder**, define the prompt caption and the prompt ID.

5. Select the source of the prompt using either the **Values from a Query** option or the **Distinct List of Values From <Field>** option.

6. Click the **Next** button.

7. Click the **Finish** button.

   The **Expression Builder** returns to focus. In the following example, the prompt was built from a distinct list of values.

8. In the **Expression Builder**, directly before the ORDER BY clause, type a WHERE clause to indicate the field on which the prompt is based and the prompt ID associated with that field. For example, if the prompt values for the City prompt are based on the values in the **State** field of the Human Resource family, you would type:

   WHERE [MI_HR_STATE_CHR] = (? :s :id= State)

   In this example, **[MI_HR_STATE_CHR]** is the field ID of the **State** field, and **(? :s :id= State)** indicates that the prompt ID of the prompt defined for the **State** field is **State**.

> **Note:** You specified this Prompt ID in step 1 when you created the State prompt.

Using the example where the values in the City prompt are filtered based on the value selected in the State prompt, the expression would look *exactly* like the following expression:

((? :s :caption='City' :id=City :query='SELECT DISTINCT [MI_HR_CITY_CHR] FROM [MI Human Resource] WHERE [MI_HR_STATE_CHR] = (? :s :id= State) ORDER BY [MI_HR_CITY_CHR])

> ŒHint: If the State prompt is configured to allow users to select multiple values, the WHERE clause syntax should be **WHERE [MI_HR_STATE_CHR] IN (? :mv :id-d=State)** instead of **WHERE [MI_HR_STATE_CHR] = (? :s :id= State)**. Notice that **=** is replaced with **IN**, and **:s** is replaced with **:mv**.

9. Click **OK** to close the **Expression Builder**.

    When you run the query, the first prompt will display a list of values according to how you specified the valid values (e.g., static list of values). The second prompt will display only values that apply to the first prompt's value. For example, if the first prompt displays a list of states, and you choose Virginia, a second prompt that displays a list of cities would display only cities that are in Virginia and that are stored in the **City** field of records where the value in the **State** field is Virginia.

Configuring a Prompt to Accept a Percent Wildcard

Query prompts can help users limit the results that will be returned by the query. In some cases, however, users might not know what to select or type in a query prompt. This might be especially true if the prompt does not present a list of values and the user must type the value as it is stored in the database in order to return any results.

To address this concern, you can configure a query prompt to accept the percent (%) wildcard. When a user types only a percent symbol in the prompt text box, the query will return records where the field on which the prompt was built contains *any* value. When a user types a combination of text and the percent symbol, the query will return records where the field on which the prompt was built contains a value that contains the specified text, preceded or followed by any combination of characters (depending on where the percent symbol is placed in the prompt text box).

To configure a prompt to accept a percent wildcard, you must use the Like operator instead of the equal (=) operator in the expression that defines the prompt.

For example, suppose you build a prompt on the Asset Manufacturer field, and the prompt is configured such that it does not present a list of values. The expression for the prompt would look something like this:

[Asset].[ASSET_MANUF_CHR] = (? :s :caption='Manufacturer' :id=Manufacturer)

If you want users to be able to type a percent symbol in the prompt to return records where the Asset Manufacturer field contains *any* value, you would need to replace the equal operator (=) with the Like operator. The complete expression would be:

[Asset].[ASSET_MANUF_CHR] Like (? :s :caption='Manufacturer' :id=Manufacturer)

When you run the query, you could type the percent symbol (**%**) in the **Manufacturer** text box, like this:

...and the results would display records where the Asset Manufacturer field contains any value, as shown in the following image.

Asset ID	Asset Manufacturer
MTR-6804	SEIMENS-AL
MTR-6721	SEIMENS-AL
MTR-7053	SEIMENS-AL
MTR-8277	SEIMENS-AL
MTR-8280	SEIMENS-AL
MTR-6647	SEIMENS-AL
MTR-8254	SEIMENS-AL
12511	SIEGER
12512	SIEGER
12518	SIEGER
12519	SIEGER

In addition, you could type **SEI%**, and the query would display records where the value in the Asset Manufacturer field begins with the letters SEI, followed by any combination of characters, such as SEIMENS-AL and SEIGER, as shown in the following image.

Asset ID	Asset Manufacturer
MTR-8198	SEIMENS-AL
MTR-6667	SEIMENS-AL
MTR-6714	SEIMENS-AL
MTR-5762	SEIMENS-AL
MTR-7288	SEIMENS-AL
MTR-8022	SEIMENS-AL
MTR-8026	SEIMENS-AL
MTR-6862	SEIMENS-AL
MTR-6863	SEIMENS-AL
MTR-6587	SEIMENS-AL
MTR-8021	SEIMENS-AL
11215	SEIGER
11217	SEIGER
11219	SEIGER

Configuring a Prompt to Return Null Values

In some cases, records might contain fields that do not contain a value. If you want to run a query with a prompt on these fields to see all records where the field is empty, you must construct the query a certain way.

This query construction is best understood through an example. The following example assumes that you want to return records where the Asset Description field of Air Cooled Heat Exchanger records is empty. It also assumes that you want to see the Asset ID, Asset Description, and Asset Installation Date of the returned records.

To configure a prompt to return records where the **Asset Description** field is empty, you would need to add the following fields to the query:

- Asset ID
- Asset Description
- Asset Installation Date
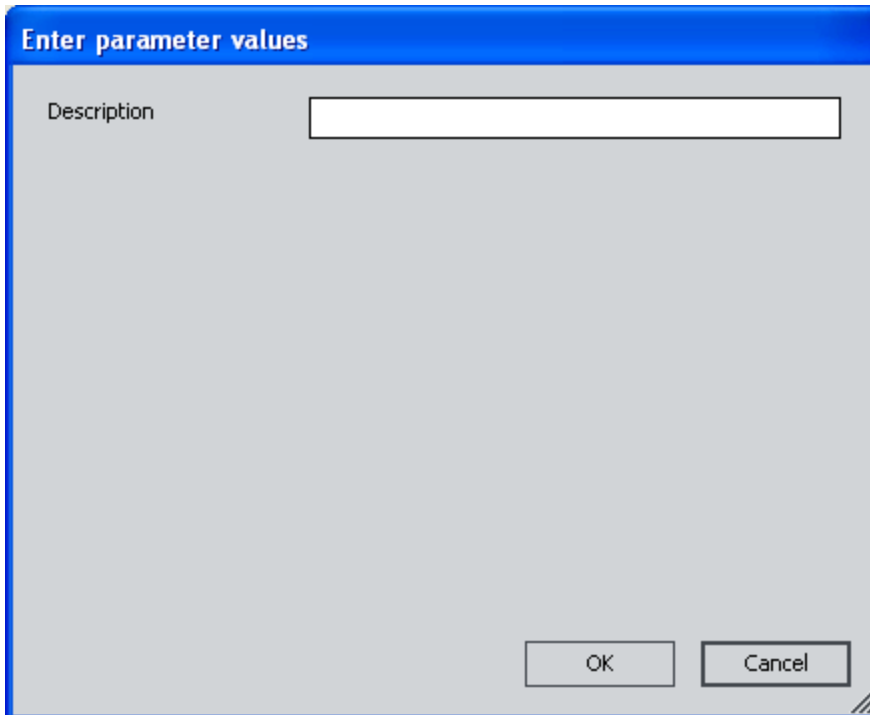
You would need to add a prompt the Asset Description field and add an expression to indicate that you want to return records where the field does not contain a value.

The design grid would look like the one shown in the following image.

Field	[Asset ID]	[Asset Description]	[Asset Installation Date]
Alias	Asset ID	Asset Description	Asset Installation Date
Table	[Air Cooled Heat Exc…	[Air Cooled Heat Exchanger]	[Air Cooled Heat Exchanger]
Sort			
Sort Index			
Include	☑	☑	☑
Display	☑	☑	☑
Hyperlink			
Criteria		(? :s :caption='Description' :id=Description)	
Or		IS NULL	

When the query is run, the following prompt would appear, as shown in the following image.

If a user clicks **OK** without providing a value in the prompt, the results will display records where the **Asset Description** field is empty, like the ones in the following image.
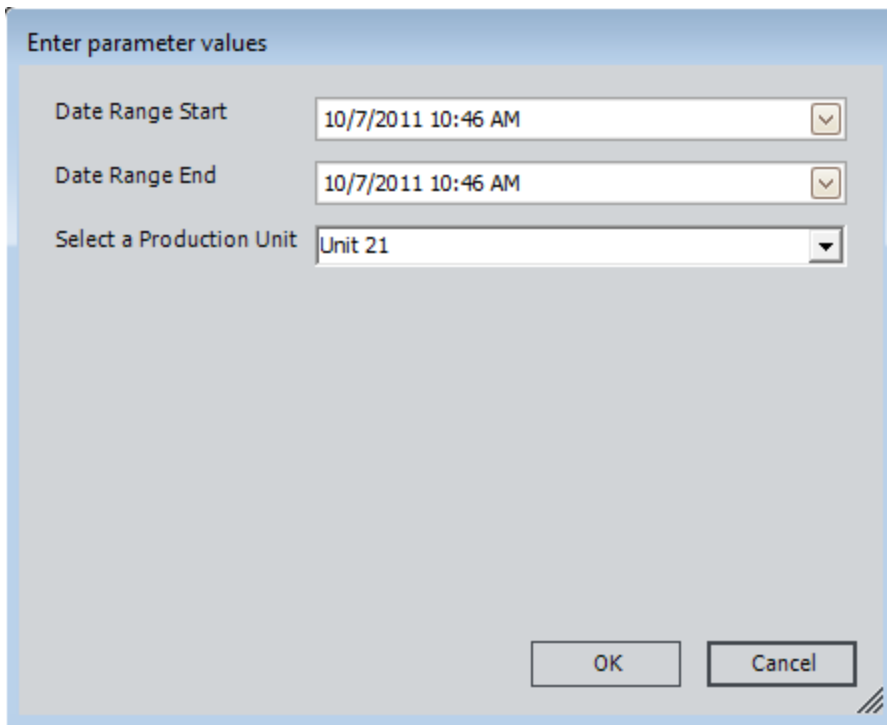
If you had not used the IS NULL operator in the query expression and a user clicked **OK** without providing a value in the prompt, the query would not return any records.

Making Prompts Appear at the Top of the Results

Suppose that your database contains a query that returns information about Production Plan records, and a prompt is defined on the query to allow you to specify the:

- Production plan start date.
- Production plan end date.
- Associated production unit.
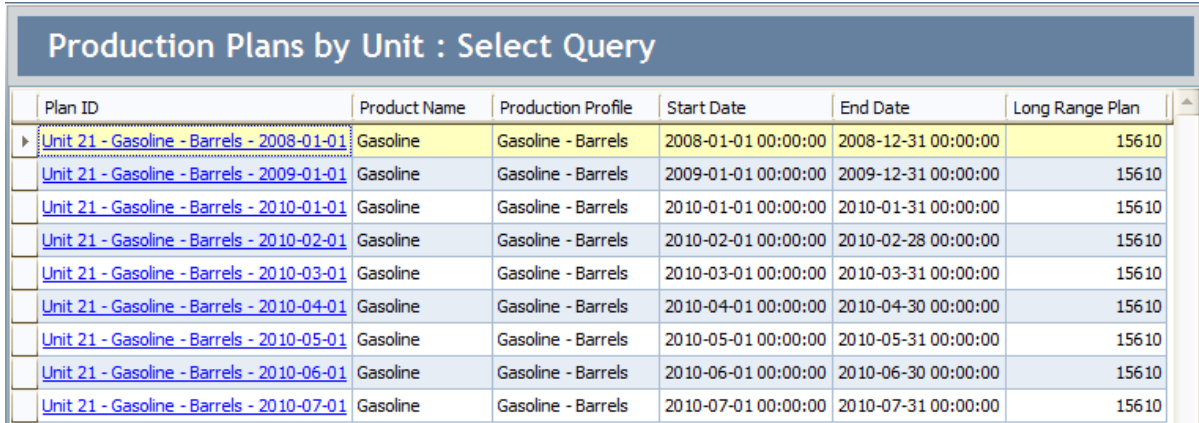
In this case, if you ran the query from the Catalog, you would see prompts in a stand-alone dialog box, like the one shown in the following image.

You would need to provide a value in each prompt and click **OK** in order to see the query results, as shown in the following image.

If you wanted the prompts to appear at the top of the results, however, you could access the query from a URL similar to the following one:

meridium://Registry/QueryBuilder?QueryPath=Public\Production Plans by Unit&Run-Only=1&promptheader=1

After you executed the URL, the results view would appear with the prompts at the top of the page, as shown in the following image.

You could select the desired values in the prompts and then click the **Run Query** button to display the results below the prompts, as shown in the following image.

Using this approach, you can continue to change the values in the prompts and rerun the query results. This approach is useful if you want to be able to see the query results along with the values that were used to yield those results.

About Prompts on Date Fields

All date fields contain both a date and time value. When you apply specific criteria to a field, you can omit the time to return all the records that contain the specified date and any time. For example, specifying the criteria =(#:D '2000-03-10') would return records where the time is 12:00:00 AM, 2:00:00 PM, 5:00:17 PM, or any other time. In other words, when you define date criteria and do not specify a time, the Meridium APM system assumes that the time is unimportant and does not take it into account when retrieving results.

In prompts, however, the time is *always* taken into account. Therefore, if you create a prompt on a date field and do not define a list of valid values, you will need to know how to enter a date that will retrieve the desired results:

- If you type a date with NO time, the query will return records where the time is 12:00 AM. Records that contain any other time will not be included in the results.

- If you type a date and a specific time, the query will return records that contain that specific date and time. Note, however, that you must specify the seconds value, but you cannot specify any value other than 00. Doing so will cause the query not to return any results.

- If you use the Calendar to select a date, you will need to modify the time manually. Either delete the time (to use a time of 12:00 AM) or type a different time. Be sure to include seconds, which cannot be typed in the Time field of the Calendar but are required in order to return results.

Because of these requirements, it may be easier always to create date prompts using the Distinct List of Values from <Field> option. This way, the list of available dates and times will always represent valid dates and times that exist in records in the database. Users will not need to know what values are stored in the database or how to enter dates in the proper format in order to return the desired results.

When you create a prompt on a date field, if that field is configured to store dates and times in UTC format (via the Configuration Manager), the time values will be displayed using the appropriate conversion from UTC to the time zone that is associated with the Security User who is logged in when the query is run. This features allows any user who runs the query to provide a local date and time in the prompt.

About Prompts on Numeric Fields

Numeric fields can have a unit of measure (UOM) associated with them. UOMs can have a conversions defined for them that cause numeric values to be converted from the stored value to a different value and UOM using UOM Conversion Sets. If a query contains a prompt on a numeric field, the associated UOM will be displayed on the **Enter parameter values** dialog box. For example, the following image shows how the UOM *Inches* is displayed for a prompt that has been defined on the Measurement Value field of the Thickness Measurement family.
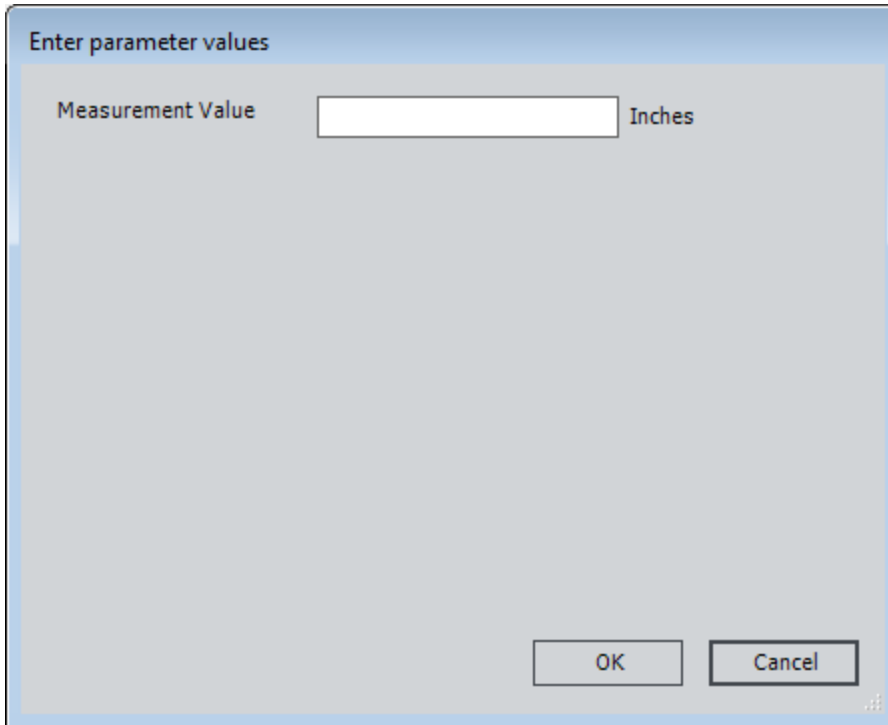
> **Note:** UOMs will be displayed only for prompts on numeric fields for which a UOM is defined. If no UOM is defined, no UOM will be displayed, and no conversion will be performed. Additionally, this behavior applies only to prompts on fields where the *stored* data type is numeric and is not converted at runtime. If a numeric field is converted to a character prompt at runtime, the UOM will not be displayed. Similarly, if a character field is converted to a numeric field at runtime, no UOM will be applied.

For queries running in *Formatted* mode, the **Enter parameter values** dialog box will display the UOM that is appropriate for you, taking into consideration the UOM Conversion Set that is associated with your Meridium APM Security User account. In order to return results, you will need to enter numeric values associated with the UOM *that is displayed*. For instance, consider the above example where the base UOM associated with the Measurement Value field is Inches, and assume that one or more records exist in the database with the value 0.25 inches the Measurement Value field.

- If you log in as a Security User with the *Meridium* UOM Conversion Set, when you run the query, the **Enter parameter values** dialog box would look like the above image. You would need to type `0.25` to return records containing the value 0.25 in the Measurement Value field.

- If you log in as a Security User with the *Metric* UOM Conversion Set, when you run the query, the **Enter parameter values** dialog box would display the UOM *Millimeters* for the Measurement Value prompt. In this case, you would need to type `6.35` (the equivalent in millimeters of 0.25 inches) to return records containing the stored value 0.25 inches. In the results, you would see the values 6.35 (millimeters) instead of 0.25 (inches).

For queries running in *Unformatted* mode, the base UOM will always be displayed, and the stored values will always be used on the **Enter parameter values** dialog box and in the query results.

Modifying Existing Prompts

If you need to modify a prompt, you should modify it in the SQL code or in the **Expression Builder**. You cannot modify a prompt using the **Prompt Builder**. Each time that you access the **Prompt Builder**, it displays the first step as if you were creating a new prompt. The **Prompt Builder** does not store settings for existing prompts.

If you want to modify an existing prompt, you can do so by editing the SQL code. Otherwise, you can recreate the prompt from scratch and included the desired modifications.

About Adding Hyperlinks to Queries

For each column in a query, you can configure one or more URLs to display hyperlinks in the query results. When a user runs the query, the Meridium APM system will build the URLs as appropriate, passing data from the query results into the URL parameters if necessary, and display in the query results hyperlinks that the user can click to access the associated feature or perform the associated function.

> **Note:** If you use variable key values in an *unformatted* query, you must include the keys as columns in the query. This rule applies to any key value. This is not a requirement for queries running in Formatted mode.

For example, you might define for a field a URL for an external website. When a user runs the query, the Meridium APM system will build a hyperlink from that URL and display the link to the user in the query results. The user will be able to click the hyperlink to open the website in Internet Explorer.

If you configure only one URL for a field, when you run the query, that URL will be used to build a single hyperlink in that cell of the query results. If you have not modified the Field cell, whatever values are found in the database for that column will serve as the text for each hyperlink. For example, if an Asset ID column contains a URL to open a record in the Record Manager, when you run the query the actual Asset IDs retrieved by the Meridium APM system will appear as hyperlinks in the results, as shown in the following image.

When only one URL has been configured, clicking the hyperlink will execute the URL behind the link. So, in this example, clicking the link would open the associated record in Record Manager.

Asset ID	Asset Description	Asset P&ID Drawing Number
HX-2521	TEMPERED WATER COOLER	3-2109-0202
HX-2522	TEMPERED WATER COOLER	3-2109-0202
HX-1798	KEROSENE COOLER	3-1951-0006
HX-1799	KEROSENE COOLER	3-1951-0006

You also have the option of configuring multiple URLs for a single column in a query. If you configure multiple URLs, they will appear on a shortcut menu when you right-click the cell in the query results, as shown in the following image. From the shortcut menu, you can click any link to execute the associated URL.

Asset ID	Asset Description
HX-2521	TEMPERED WATER COOLER
HX-252	Open in Record Manager · COOLER
HX-179	Open Task List
	View Failure Report
HX-1799	KEROSENE COOLER

When you configure multiple URLs, you can define a default URL, which is the URL that will be executed upon a left-click in the query results cell. If you configure multiple URLs

and do not define a default URL, the URL that appears *first* on the shortcut menu will be executed via a left-click.
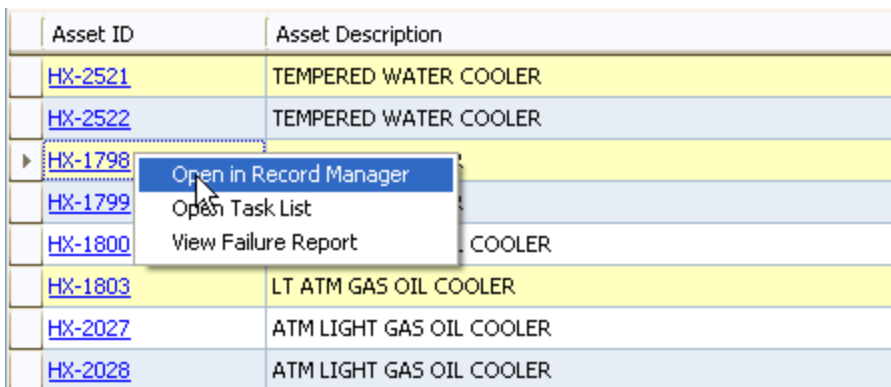
If you configure only one URL for a field, that URL will appear in the Hyperlink cell for that field in the design grid. If you configure multiple URLs, however, the Hyperlink cell will display text that indicates the number of URLs that have been configured. For example, if two hyperlinks have been configured, the text will read *<<2 hyperlinks configured.>>*.

Note that if you configure multiple URLs for a field, you should not copy the text that appears in that Hyperlink cell and paste it to another Hyperlink cell. Doing so will not copy the multiple URLs, it will copy just the text that indicates the number of the URLs that have been configured, such as *<<2 hyperlinks configured.>>*, which is not a valid URL. When you run the query, the results in the associated column will appear to be a hyperlink, but when you click the link, an error will appear.

Note also that if only one hyperlink has been configured, if you copy the URL from the hyperlink cell and paste it into another field, only the URL will be copied and pasted. If a caption has been defined for the hyperlink, that caption will not be copied and pasted.

When you configure multiple URLs, if you select multiple rows in the query results and click a link on the right-click shortcut menu, the URL will be executed for the row you selected that appears *last* in the list.

For example, in the following image, three rows are selected, and the shortcut menu has been opened in the row containing the record for HX-1798. When the URL is executed, however, it will open in the Record Manager the record represented by the last row in the list, HE-1803.

About Grouping Hyperlinks

Because you can configure multiple hyperlinks for query fields, you might want to group the hyperlinks so that similar hyperlinks appear together on submenus in the query results. For example, you might configure hyperlinks that are specific to Inspection Management and hyperlinks that provide access to external websites, such as MSN and Yahoo. To make it easier to distinguish between and find certain links, you might want to group the Inspection Management hyperlinks and the external hyperlinks onto their own separate submenus.

In the following image, the root menu contains the two submenus: **External URLs** and **Inspection URLs**. Based on the captions, we can infer the type of hyperlinks that each submenu will contain.
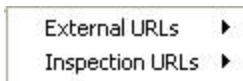
The **External URLs** submenu contains two hyperlinks: **MSN** and **Yahoo**.

The **Inspection URLs** submenu also contains two hyperlinks: **Inspection Start Page** and **Inspection Administration Page**.

You can also group hyperlinks using submenus that stem from other submenus. For example, in the following image, the Meridium APM URLs submenu contains the Inspection URLs submenu, which contains the hyperlinks **Inspection Start Page** and **Inspection Administration Page**.

You can choose how to organize links on the root menu and its submenu using the options provided on the **Menu Editor window**.

Accessing the Menu Editor Window

To access the Menu Editor window:

- In the **Hyperlink** cell of the desired query field, click the ⋯ button.

  The **Menu Editor** window appears.

Aspects of the Menu Editor Window

The **Menu Editor** window displays the hyperlinks and submenus that will appear when you right-click a cell in the query results.

The following columns are displayed:

- **Sub-Menu:** Contains a check box that, if selected, indicates that the item is a sub-menu. The submenu could be a submenu of the root menu, which appears by default when you right-click the cell, or it could be a submenu of another sub-menu.
- **Appears On:** Specifies the menu on which the submenu or hyperlink appears. This could be the default root menu or a submenu.
- **Caption:** Specifies the label that will appear to define the URL or submenu.
- **URL:** Specifies the URL behind the hyperlink. This column is empty if the item is a submenu.
- **Default:** Contains a radio button that, if selected, specifies that the URL will be launched automatically when you click hyperlinked value in the cell. This field is disabled if the item is a submenu.

From the **Menu Editor** window, you can:

- Add a new hyperlink.
- Add a new submenu.
- Modify an existing hyperlink or submenu.
- Delete a hyperlink or submenu.

Aspects of the Menu Item Information Dialog Box

The **Menu Item Information** dialog box contains the following sections:

- Hyperlink
- Sub-Menu

When you access the window by clicking the **New Menu Item** button on the **Menu Editor** window, the **Hyperlink** option is selected by default, and the fields in the **Sub-Menu** section are disabled.

The following list describes how to use each field in the **Hyperlink** section:

- **Appears On:** Select from the list the menu on which the hyperlink should appear. This could be either the root menu, which appears when you right-click the cell in the query results, or a submenu that appears on either the root menu or another submenu.
- **Caption:** Type the text that will appear as the hyperlink label when you right-click the cell in the query results. This field is required if you are creating or modifying a hyperlink.
- **URL:** Type the URL behind the hyperlink, or click the [...] button to create the URL using the **URL Builder**. This field is required if you are creating or modifying a hyperlink.
- **Make Default:** Select the check box if you want the URL to be launched automatically when you click the hyperlinked text in this cell of the query results. If you are configuring only one hyperlink, you do not need to specify that it should be the default hyperlink. It will be executed automatically when you click the hyperlinked text in the query results cell.

The following list describes how to use each field in the **Sub-Menu** section:

- **Appears On:** Select from the list the menu on which the submenu should appear. This could be either the root menu, which appears when you right-click the cell in the query results, or another submenu.
- **Caption:** Type the text that will appear as the title for the submenu. This field is required if you are creating or modifying a submenu.

You can toggle between the **Hyperlink** section and **Sub-Menu** section. The fields in the section that is not currently selected will be disabled. When you click the **Save** button, the values in the currently selected section will be saved.

Adding a New Hyperlink

To add a new hyperlink, either to the root menu or a submenu:

1. On the **Menu Editor** window, click the **New Menu Item** button.

   The **Menu Item Information** dialog box appears.

   The **Hyperlink** option is selected by default.

2. In the **Appears On** list, select the menu on which the hyperlink should appear. The default value is **[Root Menu]**, but you can select a submenu if any submenus have been created.

3. In the **Caption** text box, type a caption for the URL. The value that you type will appear as the hyperlink label when you right-click the cell in the query results. Click the ⬤ button to .

4. In the **URL** text box, type a URL for the hyperlink. Alternatively, you can click the … button to display the URL Builder, which will walk you through the process of creating a URL.

5. Select the **Make Default** check box if you want the URL to be executed automatically when you left-click the cell in the query results. Note that this option is intended to be used only if you are configuring multiple hyperlinks. If you are configuring only one hyperlink, you do not have to select the **Make Default** check box; the URL will be executed automatically.

6. Click **OK**.

   The new hyperlink appears in the list on the **Menu Editor** window.
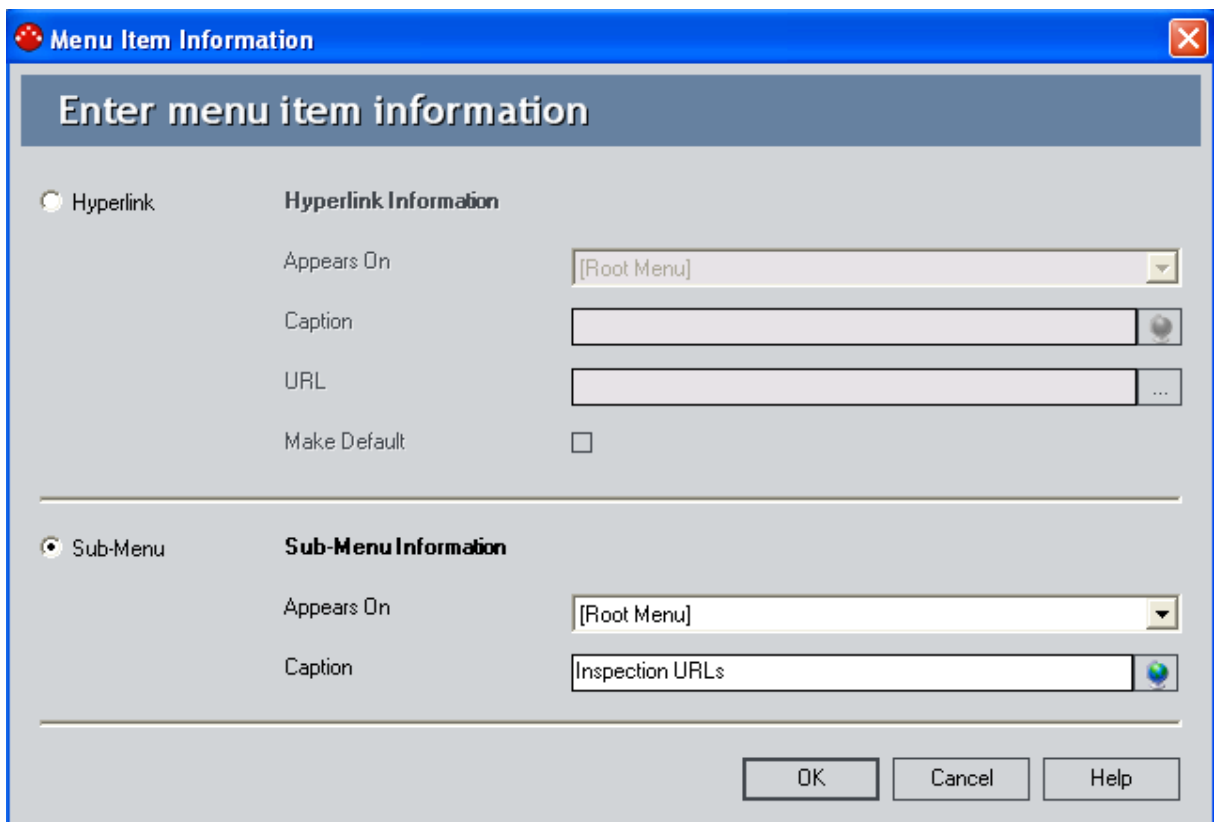
   > **Note:** To save the hyperlink with the query, you must save the query.

Adding a New Submenu

Submenus can serve as a mechanism for grouping hyperlinks of a similar type or function. The first step in configuring the submenu is to define the submenu itself, which consists of choosing a name for the submenu and specifying where it should appear. After you have completed this first step, you will need to add hyperlinks to the submenu. Note that the submenu will not appear until you have defined one or more links for it.

To add a new submenu:

1. On the Menu Editor window, click the **New Menu Item** button.

   The **Menu Item Information** dialog box appears, and the **Hyperlink** option is selected by default.

2. Select the **Sub-Menu** option.

3. In the **Appears On** list, select the menu on which the submenu should appear. The default value is **[Root Menu]**, but you can select a submenu if any submenus have been created.

4. In the **Caption** text box, type a caption for the submenu. The value that you type will appear as the submenu label. Click the ⊕ button to manage translations for that string. In the following example, the submenu caption is **Inspection URLs**.

5. Click **OK**.

    The new submenu appears in the list on the **Menu Editor** window. At this point, you can add hyperlinks to the submenu.

    > **Note:** To save the submenu with the query, you must save the query.

Editing an Existing Hyperlink or Submenu

To edit an existing hyperlink or submenu:

1. On the **Menu Editor** window, select the row containing the hyperlink or submenu that you want to modify.

2. Click the **Edit Selection** button.

   The **Menu Item Information** dialog box appears.

   If you are editing a hyperlink, the fields in the **Hyperlink** section are enabled, and the values in the **Sub-Menu Information** section are unavailable. If you are editing a submenu, the fields in the **Sub-Menu** section are enabled, and the values in the **Hyperlink Information** section are unavailable. The **Hyperlink** and **Sub-Menu** radio buttons are both disabled to indicate that you cannot toggle between them.

3. Modify the values as desired.

4. Click **OK**.

   > **Note:** To save the changes, you must save the query.

Deleting an Existing Hyperlink or Submenu

To delete an existing hyperlink or submenu:

1. On the **Menu Editor** window, highlight the row(s) containing the item(s) you want to delete.

2. Click the **Delete Selection(s)** button.

   A confirmation message appears, asking if you really want to delete the selected item(s). If you selected to delete a submenu that contains hyperlinks or submenus, you will be asked if you want to delete these hyperlinks and submenus as well. Click the **Yes** button to delete the item(s), or click the **No** button if you do not want to delete the item(s).

Linking from Query Results to the Record Manager

The simplest type of query, a Select query, retrieves information and displays it on the screen. For example, you might create a query that returns a list of all the Piping Circuit records in the database. Your results will appear in a grid that shows for each Piping Circuit record the columns that the query was configured to retrieve, as shown in the following image.

To provide access to additional information from your query results, you could create a hyperlink that opens each Piping Circuit record in the Record Manager. For example, in the following image, the contents of the Asset ID column are shown as hyperlinks.

To link a column in a query to the Record Manager, you will need to insert the following link into the **Hyperlink** cell of the column that you want to appear as the hyperlink:

meridium://Registry/RecordManager?EntityKey=$ENTY_KEY$&FamilyKey=$FMLY_KEY$

For instance, in our example, we inserted this URL in the **Hyperlink** cell of the **Asset ID** column. If we had put the URL in the **Asset Description** column, the Asset Descriptions would have appeared as hyperlinks.

> **IMPORTANT:** Note that our example contains a variable Entity Key and Family Key. If we were running the query in *Unformatted* mode, the ENTY_KEY and FMLY_KEY columns would need to be included in the SQL select statement (i.e., the **Include** check box would need to be selected), but they would not need to be displayed in the output

(i.e., the **Display** check box could be cleared). If we were running the query in *Formatted* mode, it would not need to include the ENTY_KEY and FMLY_KEY columns.

Linking from One Query to Another

Using a URL, you can link from one query to another query that provides more detailed information about the first query. Consider an example where:

- The first query displays the total cost of failures for the pump records in your database.

Motor Falure Costs : Select Query		
Asset ID	Failure ID	Total Failure Cost
MTR-0007	FAIL - 01764	7901
10080101	FAIL-1	1568.45
10080102	FAIL-35	6954.56
MTR-0087	FAIL-43	4239.14

- The second query displays for a given pump record a breakdown of the total failure cost by category, such as labor cost and material cost.

Asset ID	Total Failure Cost	Total Labor Costs	Total Material Cost
MTR-0007	7901	458	676

After you have created the two queries, you can add a hyperlink to the first query so that users will be able to access the more detailed query from the first query. The URL in the first query would be defined in the Asset ID column and might look something like this:

meridium://Registry/QueryBuilder?QueryPath=Public\Meridium\Queries\Failure Breakdown&p0=$Asset ID$

Where:

- **QueryPath=Public\Meridium\Queries\Failure Breakdown** specifies that you want to open the Failure Breakdown query (the second, more detailed query), which is stored in the Catalog folder \\Public\Meridium\Queries.

- **p0=$Asset ID$** specifies to pass values from the results of the first query to a prompt in the second query. By defining the hyperlink on the Asset ID column, Asset IDs will be passed from the results of the first query to the prompt in the second query. In this case, the second query would need to contain a prompt that accepts an Asset ID and then returns for that piece of equipment the detailed, failure cost information.

With this hyperlink defined, the results of the first query would look like those in the following image.

Asset ID	Failure ID	Total Failure Cost
MTR-0007	FAIL - 01764	7901
10080101	FAIL-1	1568.45
10080102	FAIL-35	6954.56
MTR-0087	FAIL-43	4239.14

Each Asset ID that you clicked would run the second query, which would return for that piece of equipment the detailed failure cost information. You could further extend this example by linking from the second query to a third query that shows even more detail. For example, from the second query, you could link to a query that breaks down Labor and Material Costs for the specified pump into Company, Contractor, and Outside Cost sub-categories. Then, from that query, you could link to the Record Manager to view the datasheet for the specified pump.

Note that you can also create graphs from each query. You can then add hyperlinks to the graphs so that the first graph, which displays a summary of the failure cost for all pumps, links to a second graph, which displays a breakdown of the total failure cost for a given pump.

What is an Expression?

An *expression* is a string of characters that together define a certain set of conditions to be applied to a query. In other words, an expression is the code that the Meridium APM system reads in order to determine what you want to retrieve from the database and how you want to display the query results.

You can build simple expressions that limit the query results based on criteria that is applied to a single field, or you can build complex expressions that can be used to perform calculations, reformat stored values, concatenate stored values, and so on. The possibilities of expressions are endless. As long as you understand the stored data and the way in which you want to present it to users, you can construct expressions to perform simple to very complex operations on the query data.

You can think of an expression as the combination of any of the following items that together define the conditions by which you want to limit the query results:

- Function.
- Field.
- Field value.
- Any other data required to yield a result.

For instance, consider the following expression:

TO_CHAR([Shell and Tube Heat Exchanger].[Asset Installation Date], 'yyyy')

This expression combines the TO_CHAR function, the Asset Installation Date field, and the text 'yyyy', which yields a result that is formatted as a four-digit year.

The following text is another example of an expression:

DECODE([Air Cooled Heat Exchanger].[Asset Status], 'Active', 'A', 'Inactive', 'I', 'No Status')

This expression combines the DECODE function, the Asset Status field, and additional data that indicates that a stored value of Active should return **A**, a stored value of Inactive should return **I**, and any other stored value (including null values) should return **No Status**.

Expressions can exist within SELECT statements, WHERE clauses, or HAVING clauses, or they can exist outside of these SQL components. For example, consider the following WHERE clause:

WHERE [Asset].[ASSET_MANUF_CHR] = 'GOULDS'

The expression **[Asset].[ASSET_MANUF_CHR] = 'GOULDS'** is contained within the WHERE clause.

In the design grid, you can construct an expression in the **Criteria** cell, the **Or** cell, and the **Field** cell. Expressions can be constructed manually, or they can be built with the

assistance of the **Expression Builder**, which is accessible from the **Field**, **Criteria**, and **Or** cells of the design grid.

When building expressions, keep in mind that for Units of Measure, formatted values, and translated strings, you must use the base (i.e., stored) values. For example, consider a query that contains an expression in a field that stores numeric values in inches. To filter your query results on that field, you must specify values as they are stored in that field (i.e., in inches) rather than as they are displayed (e.g., in centimeters).

For instance, to return only the records where 25 inches has been recorded in this field, your query expression must contain the value 25, not 63.5, which is the stored value converted to centimeters, which might be displayed to some users. To determine how you should construct values in your expressions, you can run the query in unformatted mode.

Why Use Expressions?

In its simplest form, a Select query displays in a grid format raw data that is stored in the database.

Consider, for example, the following query:

SELECT [MI_EQUIP000].[MI_EQUIP000_EQUIP_SHRT_DESC_C] "Equipment Short Description"
FROM [MI_EQUIP000]

This query returns the value in the Equipment Short Description field in all Equipment records. The results might look something like those shown in the following image.

Equipment Short Description
ISOFLO 2-B-1 HTR O2
2CRUDE HTR O2
2-B-1 HEATER OXYGEN
LOOP
2-B-3 HEATER OXYGEN
HEATER - NO. 2 VACUUM CHARGE
HEATER - NO. 2 CRUDE CHARGE
HEATER - PITCH
2-B-3 PILOT FLAME SCANNER
2-B-3 MAIN FLAME SCANNER
VESSEL - NO. 2 CRUDE PREFLASH TOWER
VESSEL - NO. 2 CRUDE FACTIONATOR
VESSEL - NO. 2 VACUUM TOWER
VESSEL - FOUL WATER SYSTEM
VESSEL - DIESEL STRIPPER
VESSEL - KEROSENE STRIPPER
VESSEL - CYLINDER STOCK STRIPPER TOWER
VESSEL - VACUUM OVERHEAD ACCUMULATOR
LOOP - DPRC-CRUDE MIXING VALVE
LOOP - DPRC-DP FUEL OIL / STEA

Suppose, however, that you do not want to view ALL Equipment records. Instead, you want to view only the records whose Equipment Short Description begins with the phrase *HEATER*. To accomplish this task, you could modify the query to indicate this condition:

SELECT [MI_EQUIP000].[MI_EQUIP000_EQUIP_SHRT_DESC_C] "Equipment Short Description"
FROM [MI_EQUIP000]
WHERE [MI_EQUIP000].[MI_EQUIP000_EQUIP_SHRT_DESC_C] LIKE 'HEATER%'

You can see that the first two lines of text are the same as the text in the first example query, so this query still returns Equipment Short Descriptions from Equipment records. The bold text, however, contains a WHERE clause, which identifies the criteria that will limit the query results. Notice that the WHERE clause contains an expression that uses the LIKE operator and assigns a value to it: HEATER%. This syntax means that you want to return only Equipment Short Descriptions that begin with the letters HEATER, followed by any combination of letters or numbers.

The results of this query might look something like those shown in the following image.

Equipment Short Description
HEATER - NO. 2 VACUUM CHARGE
HEATER - NO. 2 CRUDE CHARGE
HEATER - PITCH
HEATER (B-7) CONVECT SECTION
HEATER (B-8) CONVECT SECTION
HEATER - GUARD CASE FEED PREHEATER
HEATER - GUARD CASE FEED PREHEATER
HEATER (B-7) STACK
HEATER (B-7) CNVECTN SECTION
HEATER - TANK 67 BURNER
HEATER - FOR PITCH TANK #98
HEATER - PITCH TANK 70
HEATER - PITCH TANK 71

The query manipulation explained above causes the query to return raw data stored in the database based on a set of conditions that you define. You can also manipulate query results by modifying the format of the raw data or performing calculations on stored values and returning the results of those calculations. This type of data manipulation is accomplished using *functions*.

About Expression Syntax

When you create an expression, a specific syntax is required in order to run the query. This documentation does not explain the syntax you should use in every possible expression. It does, however, explain how you should format field values in expressions on the following field types:

- Character fields
- Text fields
- Date fields
- Logical fields
- Numeric fields

Formatting Expressions on Character Fields

When you create an expression on a character field, the actual field value(s) must be surrounded by single quotation marks. For example, if you want to return records where the Asset Status field contains the value *Active*, the syntax would be:

WHERE [Asset].[ASSET_STAT_CHR] = 'Active'

...where **Active** is surrounded by single quotation marks because it is the stored value that you want to use for limiting the query results.

If you want to search on multiple values, such as *Centrifugal Pump* and *Rotating Pump*, you would again insert single quotation marks around the actual values. The syntax would be:

WHERE ([Asset].[ASSET_TYPE_CHR] = 'Centrifugal Pump' AND [Asset].[ASSET_TYPE_CHR] = 'Rotating Pump')

...where **Centrifugal Pump** and **Rotating Pump** are surrounded by single quotation marks because they are the actual values to be used in the query criteria.

> **Note:** When you are using an Oracle schema, the value is case sensitive by default. This means, for example, that if you type **WHERE [Asset].[ASSET_STAT_CHR] = 'ACTIVE'** for a field where values are stored in the database as **Active**, the query will not return any results. You can turn this case-sensitive option off via the **Query Tasks** menu. This option is available only on Oracle schemas. On a SQL Server database, query expression values are not case sensitive.

Formatting Expressions on Text Fields

When you create an expression on a text field, the value within the expression must be surrounded by single quotation marks. For example, if you want to create an expression for the Asset Description field to return the records where the field contains the value *This is a test*, the expression would be:

WHERE [Asset].[ASSET_DESC_CHR] LIKE 'This is a test'

...where **This is a test** is surrounded by single quotation marks because it is the stored value that you want to use to limit the query results.

Note that text fields are stored in the database differently than character fields and are, therefore, handled differently. One difference is that you cannot use the = operator with text fields. Instead, you must use the like operator. *When creating text field expressions, be sure to specify the like operator.*

> **Note:** When you are using an Oracle schema, the value is case sensitive by default. This means, for example, that if you type **WHERE [Asset].[ASSET_DESC_CHR] LIKE 'THIS IS A TEST'** in a field where values are stored in the database as **This is a test**, the query will not return any results. You can turn this case-sensitive option off via the **Query Tasks** menu. This option is available only on Oracle schemas. On a SQL Server database, query expression values are not case sensitive.

Formatting Expressions on Date Fields

The syntax you should use to construct expressions on date fields is:

(# :D 'yyyy-mm-dd')

> **Note:** You will need to use the syntax yyyy-mm-dd on ALL workstations, regardless of your Meridium APM Culture setting or your Windows Regional and Language options. This means that even if dates are displayed to you in month/day/year format, the expression should still use the yyyy-mm-dd format.

For example, if you wanted to create an expression on the Asset Installation Date field to return the records where the field contains the value 01-03-1994 (where 01 represents the first month and 03 represents the third day of the month), the expression would be:

WHERE [Asset].[ASSET_INSTL_DT] = (# :d '1994-01-03')

Formatting Expressions on Logical Fields

A logical field represents a value of True or False. When you create an expression for a logical field, the exact syntax to use is either **'Y'** (True) or **'N'** (False).

For example, if you want to create an expression on the Spared field to return records where the check box is selected (the value is True), the expression would be:

WHERE [Asset].[ASSET_SRPD_IND] = 'Y'

Formatting Expressions on Numeric Fields

When you create an expression for a numeric field, you must use the exact numeric values that are stored in the database. Some numeric values may be displayed differently than they are stored. For example, numbers may be formatted to display a certain number of decimal places, to include a currency symbol, or to be converted to a different unit of measure (UOM).

To determine how values are stored in the database, run the query in unformatted mode. Then, format numeric values in your query expressions exactly as they are displayed in the unformatted query results.

Unlike character and text field expression values, which require single quotation marks, the actual numeric values that you want to return do not need to be formatted in a particular way in your expression. Specifying the value as it stored is sufficient to return the expected results. For example, an acceptable way to express that returned records should contain a value of 5 in the Number of Storage Tanks field is:

WHERE [MI_CRITANAL].[MI_CRITANAL_SYS_NO_TANKS_N] = 5

Constructing an Expression in the Field Cell

If you want to add an expression to your query, you can type it in the **Field** cell of the design grid. If desired, you can use the **Field** cell to write expressions involving fields that are already included in the results. For instance, suppose you want to display an installation date in two formats: the stored format and a modified format. You could add the Installation Date field once to return the date as it is stored, and you could write a DECODE statement in a separate **Field** cell to indicate how you want to reformat the date. If you select the **Display** check box for each column, the query results will display the installation date twice.

You can add an expression to the **Field** cell by:

- Typing the expression manually.
- Accessing the **Expression Builder** from the **Field** cell.

If you add an expression to the **Field** cell, you must type the exact syntax that is required in order to run the query. The Meridium APM system will not reformat any text that you type in a **Field** cell.

Constructing an Expression in the Criteria Cell

If you want to add an expression to your query, you can type it in the **Criteria** cell of the design grid. You can use the **Criteria** cell of the design grid to add a condition to the query to limit the results. Note that anything you add to the **Criteria** cell will be pre-pended with the text WHERE or HAVING in the SQL code. The text WHERE and HAVING is not displayed in the design grid.

For example, suppose you want to return pieces of equipment that were manufactured by GOULDS. You could add the Asset Manufacturer field to the query and use the **Criteria** cell to limit the results to only those pieces of equipment where the Asset Manufacturer field contains the value *GOULDS*. The design grid would look something like the following image.

Field	[Asset Manufacturer] ▼ ⋯	[Asset ID]
Alias	Asset Manufacturer	Asset ID
Table	[Asset]	[Asset]
Sort		
Sort Index		
Include	☑	☑
Display	☑	☑
Hyperlink		
Criteria	('GOULDS')	
Or		

Notice that *('GOULDS')* appears in the **Criteria** cell. While this text alone does not constitute an entire expression, Meridium APM interprets the data in the remaining cells and constructs a valid expression. For example, the **Field** cell indicates the field from which you want to retrieve data, and the **Table** cell indicates the table from which you want to retrieve data. Likewise, the equal (=) operator is not specified, but it is understood. The text WHERE is also unspecified but understood by Meridium APM logic. If you were to view the SQL code, you would see this:

SELECT [Asset].[ASSET_MANUF_CHR] "Asset Manufacturer", [Asset].[ASSET_ID_CHR]
"Asset ID"
FROM [Asset] JOIN_SUCC [Failure] ON {Asset Has Failure}
WHERE [Asset].[ASSET_MANUF_CHR] = 'GOULDS'

You can see that the expression *[Asset].[ASSET_MANUF_CHR]= 'GOULDS'* has been inserted automatically into the WHERE clause at the end of the code.

Note that when you construct an expression to the **Criteria** cell, you can type the exact syntax that is required to run the query or you can type something that is close to the required syntax and let the Meridium APM system reformat it automatically.

Constructing an Expression in the Or Cell

If you want to add an expression to your query, you can type it in the **Or** cell of the design grid. You can use the **Or** cell of the design grid to add a condition to the query to limit the results. Note that anything you add to the **Or** cell will be pre-pended with the text WHERE or HAVING in the SQL code. The text WHERE and HAVING is not displayed in the design grid.

For example, suppose you want to return Equipment records that were manufactured by JENSEN or WESTERN SUPPLY. You could add the Manufacturer field to the query and use the **Criteria** cell and the **Or** cell to limit the results to only those Equipment records where the Manufacturer field contains the value *JENSEN* or the value *WESTERN SUPPLY*. The design grid would look something like the following image.

Field	[Manufacturer]	[Equipment Short Description]
Alias	Manufacturer	Equipment Short Description
Table	[Equipment]	[Equipment]
Sort		
Sort Index		
Include	☑	☑
Display	☑	☑
Hyperlink		
Criteria	'JENSEN'	
Or	'WESTERN SUPPLY'	

Notice that *'JENSEN'* appears in the **Criteria** cell and *'WESTERN SUPPLY'* appears in the **Or** cell. While this text alone does not constitute an entire expression, Meridium APM interprets the data in the remaining cells and constructs a valid expression. For example, the **Field** cell indicates the field from which you want to retrieve data, and the **Table** cell indicates the table from which you want to retrieve data. Likewise, the equal (=) operator is not specified, but it is understood. The text WHERE is also unspecified but understood by Meridium APM logic. If you were to view the SQL code, you would see this:

SELECT [MI_EQUIP000].[MI_EQUIP000_MFR_C] "Manufacturer", [MI_EQUIP000].[MI_
EQUIP000_EQUIP_SHRT_DESC_C] "Equipment Short Description"
FROM [MI_EQUIP000]
WHERE ([MI_EQUIP000].[MI_EQUIP000_MFR_C] = 'JENSEN' OR [MI_EQUIP000].[MI_
EQUIP000_MFR_C] = 'WESTERN SUPPLY')

You can see that the following expression has been inserted automatically into the WHERE clause at the end of the code:

([MI_EQUIP000].[MI_EQUIP000_MFR_C] = 'JENSEN' OR [MI_EQUIP000].[MI_EQUIP000_
MFR_C] = 'WESTERN SUPPLY')

Note that when you add an expression to the **Or** cell, you can type the exact syntax that is required to run the query or you can type something that is close to the required syntax and let the Meridium APM system reformat it automatically.

Using the previous example, the query results might look something like the following image.

Manufacturer	△	Equipment Short Description
JENSEN		MIXER - TANK 803
JENSEN		MIXER - TANK 804
WESTERN SUPPLY		EXCHANGER - KEROSENE COOLER
WESTERN SUPPLY		EXCHANGER - KEROSENE COOLER
WESTERN SUPPLY		EXCHANGER - DIESEL COOLER
WESTERN SUPPLY		EXCHANGER - DESALTED CRUDE VS DIESEL
WESTERN SUPPLY		EXCHANGER - LSRN CONDENSER

Aspects of the Expression Builder

The **Expression Builder** contains various fields and controls to assist you in constructing an expression for your query criteria. The **Expression Builder** window is divided into five main sections, as shown in the following labeled i. We use these labels later to refer to the various parts of the window.
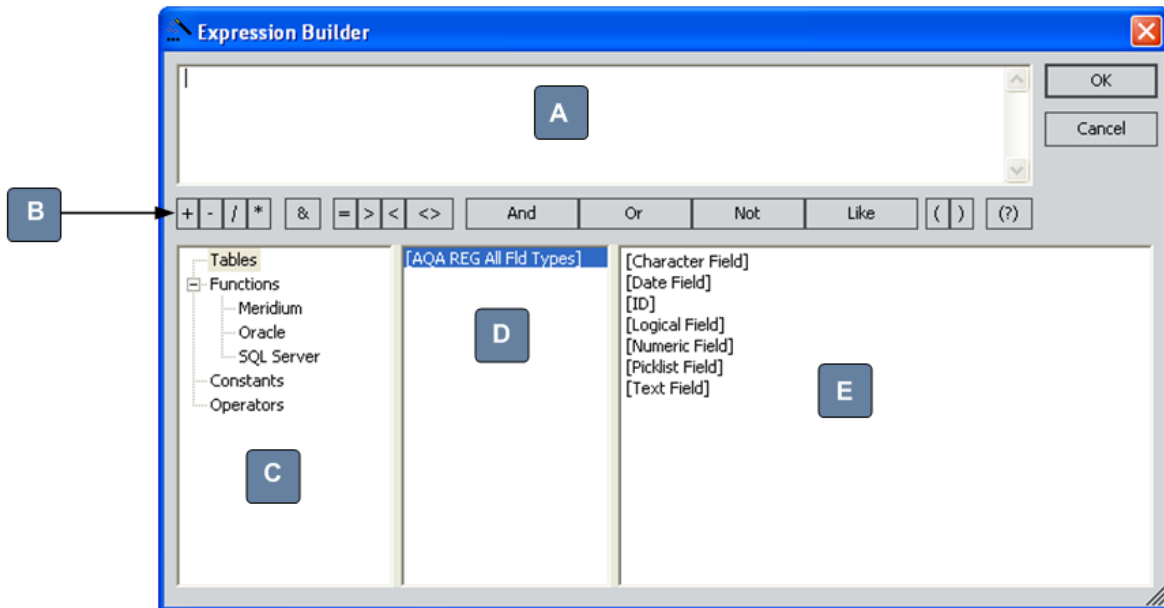
A. A text box that displays the expression itself. You can type directly in this text box or insert values by clicking buttons and selecting fields in the window, which causes the expression to be built dynamically as a result of the values you select.

B. Buttons that insert symbols to establish a relationship between or group parts of an expression. You can click a button to insert the corresponding symbol into the expression.

> **Note:** The button lets you add a prompt to a query.

C. A list that displays the categories for the valid components, or parts, of the expression. The expression can include:

- **System Codes:** The System Codes that you can include in the expression. This option appears *only* if a System Code Table Valid Values list has been defined via the Configuration Manager for the selected field (i.e., if no Valid Values list has been defined, if the Valid Values list uses a static list of values, or if the Valid Values list is By Rule, this option does not appear). If this option appears and you select it, list **E** displays:

  - ALL the codes and descriptions for the appropriate System Code Table *if* the Valid Values rule is defined as **System Code Table Only.**

  - ONLY the codes and descriptions that meet the literal value criteria *if* the Valid Values rule is defined as **System Code Table with Literal Reference**. Additionally, if a literal reference has been defined, the reference value will appear in parentheses beside the table ID in list **D.**

  - ALL the codes and descriptions for the appropriate System Code Table *if* the Valid

Values rule is defined as **System Code with Field Reference** (i.e., in the absence of a field value to use as the reference, there is no way to limit the list).

> **Note:**When you select a System Code to include in your expression, the System Code ID (rather than the description) will be used. Even though System Code descriptions appear in the datasheet, the IDs are actually stored in the database. By using the ID instead of the description in the expression, your criteria will apply to ALL System Codes, including those with different, translated descriptions.

- **Tables:** The database tables from which you can select fields to include in the expression. When you select the **Tables** option in list **C**, list **D** displays the query source families, and list **E** displays all the fields in those families.

- **Functions:** The three types of functions that you can include in the expression: **Meridium APM**, **Oracle**, and **SQL Server**. When you select a function category in list **C**, list **D** displays the types of functions in that category, and list **E** displays all the available functions.

- **Constants:** The constants that you can include in the expression. A constant is a static value provided for comparative reasons. When you select the **Constants** option in list **C**, list **D** displays the query source families, and list **E** displays all the fields in those families.

- **Operators:** The symbols that you use to join parts of the expression. When you select the **Operators** option in list **C**, list **D** displays different classifications of operators, and list **E** displays all the valid operators.

D. A list that displays a subset of options, based on your selection in list **C**. You can select an item in this list to limit the options in list **E**.

E. A list that displays a subset of values based on your selections in lists **C** and **D**. You can double-click any item in the list to insert that value into the expression. The value will be inserted wherever the cursor is currently positioned.

Creating an Expression Using the Expression Builder

> **Note:** When you construct expressions in the **Expression Builder**, the system does not check whether your syntax is correct. The **Expression Builder** simply provides an interface for selecting the criteria to include in the expression.

To define criteria for a field via the Expression Builder:

1. In the Meridium APM Framework, for the query in which you want to create an expression, access the query design.

2. In the appropriate **Field**, **Criteria**, or **Or** cell of the design grid, click the ⋯ button.

   The **Expression Builder** appears.
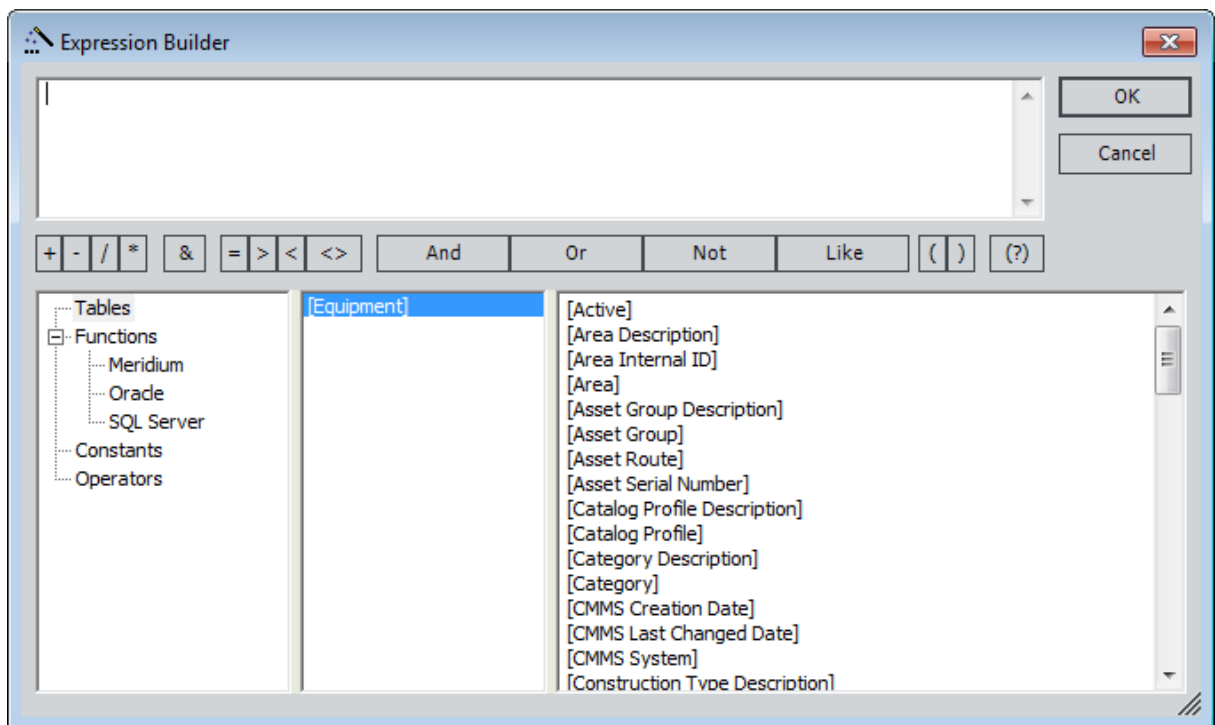
3. Using the fields provided, construct the desired expression.

4. Click **OK**.

   The expression you constructed appears in the cell.

About Automatic Reformatting

If you construct an expression in the **Criteria** cell or the **Or** cell of the design grid, the Meridium APM system will sometimes reformat it automatically. This automatic reformatting functionality is intended to save you time so that you are not required to type the exact syntax required in order to run the query. In some cases, however, the reformatted text might not yield results, depending on the stored values and the reformatted syntax.

When you type something in the **Criteria** cell or **the** Or cell, the Meridium APM system will reformat it if the field type is:

- Character
- Text
- Date
- Logical

How Meridium APM Reformats Expressions on Character Fields

If you type something in the **Criteria** cell or the **Or** cell of the design grid for a character field and you do not type single quotation marks yourself, the Meridium APM system will insert them automatically for you. For example, if you want to search for records where the Asset Status is Active, you could type either `'Active'` or `Active`. If you typed `Active`, the Meridium APM system would insert the single quotation marks automatically, reformatting the typed text as **'Active'**.

If the expression contains multiple words or if you want to use operators other than equals and you do not type the single quotation marks yourself, the Meridium APM system will insert the single quotation marks around the *entire* phrase, including the operators. Depending upon the values stored in the database, this may or may not return the desired results.

For example, suppose you create a query on the Shell and Tube Heat Exchanger family, and you add the Asset Manufacturer field to the design grid. If there are five different manufacturers of Shell and Tube Heat Exchangers, but you want to return Shell and Tube Heat Exchangers manufactured by only two of those manufacturers, *Alco* and *Whitlock*, the query expression on the Manufacturer field must be formatted as follows:

='Alco' or 'Whitlock'

If you type `Alco or Whitlock` in the **Criteria** cell for the Asset Manufacturer field, the Meridium APM system will reformat the expression as **'Alco or Whitlock'**. The query will not return any results because the syntax suggests that you want to find Shell and Tube Heat Exchangers whose manufacturer is *Alco or Whitlock*, which does not exist in the database.

Likewise, if you type `=Alco or Whitlock`, the Meridium APM system will reformat the expression as **'=Alco or Whitlock'**, which, again, would not return any results because there is no manufacturer named *=Alco or Whitlock* in the database.

To return the desired results, you should either type `'Alco' or 'Whitlock'` in the **Criteria** cell or type `Alco` in the **Criteria** cell and `Whitlock` in the **Or** cell.

> **Note:** If type an entirely numeric value (e.g., 123) or an expression that can be interpreted as a mathematical equation (e.g., 123-12), the Meridium APM system will not insert quotation marks automatically. This will cause the expression to be invalid, and when you run the query, the system will display an error. To resolve this problem, add the quotation marks manually.

How Meridium APM Reformats Expressions on Text Fields

If you type something in the **Criteria** cell or the **Or** cell of the design grid for a text field and you do not type single quotation marks yourself, the Meridium APM system will insert them automatically for you. For example, if you want to search for records where the Asset Additional Information field contains the text *This asset exists for testing purposes*, you could type the phrase with or without the single quotation marks. If you omitted the single quotation marks, the Meridium APM system would insert them for you.

Note that while the Meridium APM system will insert the single quotation marks automatically, it will not insert the like operator. *When creating text field expressions, be sure to specify the like operator.* Otherwise, the Meridium APM system will assume the = operator and return an error.

How Meridium APM Reformats Expressions on Date Fields

The Meridium APM system is configured to format dates according to your Meridium APM Culture setting and your Windows Regional and Language options. Format rules on date fields can override these settings, however. To determine how the Meridium APM system will interpret dates that you type in your expressions (i.e., the order of the numbers), you should run the query in *unformatted* mode to see how the dates are displayed. When you type an expressions in the **Criteria** cell or the **Or** cell for a date field, then, you should format dates however they are displayed to you in the results when you run the query in unformatted mode (i.e., type your date figures in the order in which they appear in the query results).

For example, if dates are displayed to you using a Day/Month/Year format, the Meridium APM system will assume that the numbers you type in your query expression follow the same Day/Month/Year format. If you type **4/5/2005**, and dates are displayed to you as Day/Month/Year, the Meridium APM system will assume that you are indicating the fourth day of May, 2005. Likewise, if you type **4/5/2005**, and dates are displayed to you as Month/Day/Year, the Meridium APM system will assume that you are indicating the fifth day of April, 2005.

Some of the accepted formats that will be reformatted are in the following list. For the purpose of these instructions, we assume that your dates are formatted as Month/Day/Year and the expression is meant to return January 05, 1997.

Common Accepted Formats

- 1/5/1997
- 1-5-1997
- 01/05/1997
- 01-05-1997
- 1 5 1997
- 1 5 97

> **Note:** If you want to specify a time along with the date, you will need to type it yourself. When the Meridium APM system reformats a date that you type, it will not insert a time, in which case, the query results will include records with that date *and any time*. If you want to query on a specific date and time, use the syntax **(# :dt 'YYYY-MM-DD hh:mm:ss')**. For example, 1/1/2006 5:00 PM would be formatted as **(# :dt '2006-01-01 17:00:00')**.

How Meridium APM Reformats Expressions on Logical Fields

Some common syntax options that the Meridium APM system will reformat to match the syntax required for logical fields are:

- **True** (using any case combination, such as true or TRUE)
- **False** (using any case combination, such as false or FALSE)
- **Yes** (using any case combination, such as yes or YES)
- **No** (using any case combination, such as no or NO)

What is a Function?

A *function* is a SQL component that manipulates data and returns a value that is not stored in the database but is derived from calculating or reformatting values. Functions can be used to calculate or reformat values stored in the database, or they can be used to calculate or format values based on static data such as the current date.

In SQL code, a function can be included as part of the SELECT statement, WHERE clause, or HAVING clause, or it can exist outside of these SQL components.

For example, in the following SQL code, the SUM function is displayed in bold text.

SELECT [Asset].[ASSET_ID_CHR] "Asset ID", **Sum([Failure].[EFAIL_TOTCST_FROM]) "Total Failure Cost"**
FROM [Asset] JOIN_SUCC [Failure] ON {Asset Has Failure}
WHERE [Failure].[EFAIL_TOTCST_FRM] > 50000
GROUP BY [Asset].[ASSET_ID_CHR]

In this example, the SUM function is contained within the SELECT statement.

Likewise, in the following example, the SUM function appears twice: once in the SELECT statement and once in the HAVING clause.

SELECT [Asset].[ASSET_ID_CHR] "Asset ID", **Sum([Failure].[EFAIL_TOTCST_FROM]) "Total Failure Cost"**
FROM [Asset] JOIN_SUCC [Failure] ON {Asset Has Failure}
GROUP BY [Asset].[ASSET_ID_CHR]
HAVING **Sum([Failure].[EFAIL_TOTCST_FROM]) > 50000**

> **Note:** Throughout this documentation, functions are grouped into categories according to how they are grouped in the Expression Builder.

GROUP BY Function

The GROUP BY function is used to group query results. For instance, suppose you want to see the total number of failures your equipment has experienced per equipment manufacturer. You might construct a query like the one shown in the following image.

In this example, the Asset Manufacturer field and the Failure ID field have been added to the design grid. The **Total** row indicates that you want to group the results by manufacturer and display a total count of failures for each manufacturer that is returned. The sort preference indicates that you want to sort the results in descending order according to the failure count.

The results might look something like those shown in the following image.

Asset Manufacturer	Failure ID
▶ INGERSOL-D	407
BW/IP	201
SLAGLE MFG	105
GOULDS	76
INDUS FABR	72
SUNDYNE	45
BERLIN CHA	41
BINGHAM	35

From these results, you can see that each row represents a different manufacturer. In other words, the same manufacturer does not appear twice because the query is grouped by the manufacturer.

In this example, the query results are grouped by one field only. You can, however, group query results by more than one field. Keep in mind that the more fields you group by, the more results you will see. When you group a query by multiple rows, the query determines all possible combinations of results and returns each distinct combination. Because we grouped the query by only one field, Asset Manufacturer, each row could return one and only one manufacturer.

If we had added the Asset ID field to the query, the results would have contained many more rows because there are more combinations to display. For instance, in the following image, you can see that the manufacturer BW/IP is displayed twice because BW/IP manufactures multiple equipment items: PMP-5382 and PMP-4543.

Asset Manufacturer	Failure ID	Asset ID
▶ BW/IP	32	PMP-5382
BW/IP	30	PMP-4543
INGERSOL-D	20	PMP-4017
WHITLOCK	20	HE-1907
INDUS FABR	19	HE-2131
GOULDS	19	PMP-5707

What Are Aggregate Functions?

When a query contains an aggregate function, the Meridium APM system performs a calculation on multiple values in a single field across multiple records and returns a single value based on that calculation. Throughout this documentation, we refer to queries that contain aggregate functions as *aggregate queries*.

You can select the following aggregate functions in the **Total** cell of the design grid:

- SUM (Sum)
- AVG (Average)
- MIN (Minimum)
- MAX (Maximum)
- COUNT (Count)
- STDEV (Standard Deviation)

The SUM and AVERAGE functions can be used only on numeric query fields. The MIN, MAX, and COUNT functions can be used on date, numeric, or character fields.

Aggregate functions must be used in conjunction with the GROUP BY function, which specifies how the query results will be grouped and displayed. This means that every field included in an aggregate query must use either the GROUP BY function or an aggregate function. In other words, if you use an aggregate function on any field in a query, another field in the same query must contain the GROUP BY function.

Available Aggregate Functions

The following table provides a list of all of the aggregate functions that you can use in queries. The **Oracle Function** column contains the function you will need to use if you are using an Oracle schema. The **SQL Server Function** column contains the function you will need to use if you are using a SQL Server database.

Description	Oracle Function	SQL Server Function
Average	AVG	AVG
Count	COUNT	COUNT
Maximum	MAX	MAX
Minimum	MIN	MIN
Standard Deviation	STDV	STDV
Sum	SUM	SUM
Variance	VARIANCE	VAR

SUM

Suppose you own a manufacturing plant, and you want to view the total salary for all of your employees, grouped by employee title. You know that you pay a salary to managers, maintenance workers, inspectors, and operators.

In this case, you might configure a query like the one shown in the following image.
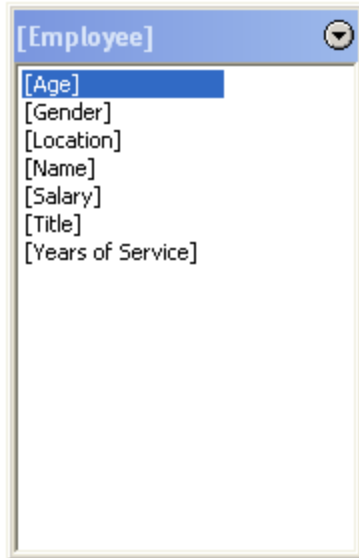
Field	[Salary] ▼ ⋯	[Title]
Alias	Salary	Title
Table	[Employee]	[Employee]
Total	Sum	Group By
Sort		
Sort Index		
Include	☑	☑
Display	☑	☑
Hyperlink		
Criteria		
Or		

In this example, to return the total salary for all employees, grouped by employee title, you would select to show the totals (via the **Query Tasks** menu), and then you would select **Sum** for the **Salary** field and **Group By** for the **Title** field. Doing so will group the results by employee title and indicate the total salary for all employees with each title.

The results might look something like those shown in the following image.

Salary	Title
70000	Inspector
114500	Maintenance Worker
90000	Manager
325500	Operator

From these results, you can see that you are paying a total of $70,000 per year for all inspectors, $114,000 per year for all maintenance workers, and so on. If you had a longer list of titles, you could sort the results by salary so you could see at a glance where the majority of the salary dollars are going.

Note that the salary figures in these results do not indicate the job titles that earn the highest salary on an *individual* basis. For example, although you are paying $325,500 per year for all operators, operators are not the highest paid positions in your business. In fact, managers earn the highest salary. You simply have more operators than you have managers, so the *total* salary that you pay to all operators is higher than the total salary that you pay to all managers.

To see the number of employees with each job title, you could add the Title field to the query a second time and use the COUNT function. Doing so would yield results like those shown in the following image.

Salary	Title	# of Employees with Title
70000	Inspector	2
114500	Maintenance Worker	4
90000	Manager	2
325500	Operator	5

From these results, you can see that you have five operators and only two managers, which is why you are paying more for your operators than you are paying for your managers.

> **Note:** The $/year unit of measure is not displayed in the results regardless of the output mode, so you would need to be familiar with your data to know that each figure represents a salary of dollars per hour.

Other Possible Uses

- Total failure cost associated with specific equipment items.
- Total maintenance cost associated with specific manufacturers.
- Total amount of downtime for specific pieces of equipment.

AVG

Suppose you own a manufacturing plant, and you want to know the average salary for all of your employees, grouped by employee title. You know that you pay a salary to managers, maintenance workers, inspectors, and operators.

In this case, you might configure a query like the one shown in the following image.

In this example, to return the average salary for all employees, grouped by employee title, you would select to show the totals (via the **Query Tasks** menu), and you would select **Avg** for the **Salary** field and **Group By** for the **Title** field. Doing so will group the results by employee title and indicate the average salary for all employees with each title.

The results might look something like those shown in the following image.

Salary	Title
35000	Inspector
28625	Maintenance Worker
45000	Manager
65100	Operator

From these results, you can see that inspectors make an average of $35,000 per year, maintenance workers make an average of $28,625 per year, and so on.

> **Note:** The $/year unit of measure is not displayed in the results regardless of the output mode, so you would need to be familiar with your data to know that each figure represents a salary of dollars per hour.

Other Possible Uses

- Average failure cost for all failures associated with a specific piece of equipment.
- Average failure cost associated with a specific type of failure.
- Average number of failures during a daylight shift versus the average number of failures during a night shift.

MIN

Suppose you own a manufacturing plant, and you want to know the minimum salary for all of your employees, grouped by employee title. You know that you pay a salary to managers, maintenance workers, inspectors, and operators.

In this case, you might configure a query like the one shown in the following image.

Field	[Salary]	[Title]
Alias	Salary	Title
Table	[Employee]	[Employee]
Total	Min	Group By
Sort		
Sort Index		
Include	☑	☑
Display	☑	☑
Hyperlink		
Criteria		
Or		

In this example, to return the minimum salary for all employees, grouped by employee title, you would select to show the totals (via the **Query Tasks** menu), and you would select **Min** for the **Salary** field and **Group By** for the **Title** field. Doing so will group the results by employee title and indicate the minimum salary for all employees with each title.

The results might look something like those shown in the following image.

MIN

Salary	Title
▶ 35000	Inspector
25000	Maintenance Worker
45000	Manager
20000	Operator

From these results, you can see that inspectors make a minimum of $35,000 per year, maintenance workers make a minimum of $25,000 per year, and so on.

> **Note:** The $/year unit of measure is not displayed in the results regardless of the output mode, so you would need to be familiar with your data to know that each figure represents a salary of dollars per hour.

Other Possible Uses

- Minimum failure cost for all failures associated with a specific piece of equipment.
- Minimum failure cost associated with a specific type of failure.
- Minimum number of failures during a daylight shift versus the minimum number of failures during a night shift.

MAX

Suppose you own a manufacturing plant, and you want to know the maximum salary for all of your employees, grouped by employee title. You know that you pay a salary to managers, maintenance workers, inspectors, and operators.

In this case, you might configure a query like the one shown in the following image.

Field	[Salary]	[Title]
Alias	Salary	Title
Table	[Employee]	[Employee]
Total	Max	Group By
Sort		
Sort Index		
Include	☑	☑
Display	☑	☑
Hyperlink		
Criteria		
Or		

In this example, to return the maximum salary for all employees, grouped by employee title, you would select to show the totals (via the **Query Tasks** menu), and you would select **Max** for the **Salary** field and **Group By** for the **Title** field. Doing so will group the results by employee title and indicate the maximum salary for all employees with each title.

The results might look something like those shown in the following image.

Salary	Title
▶ 35000	Inspector
31000	Maintenance Worker
45000	Manager
25000	Operator

From these results, you can see that inspectors make a maximum of $35,000 per hour, maintenance workers make a maximum of $31,000 per hour, and so on.

> **Note:** The $/year unit of measure is not displayed in the results regardless of the output mode, so you would need to be familiar with your data to know that each figure represents a salary of dollars per hour.

Other Possible Uses

- Maximum failure cost for all failures associated with a specific piece of equipment.
- Maximum failure cost associated with a specific type of failure.
- Maximum number of failures during a daylight shift versus the maximum number of failures during a night shift.

COUNT

Suppose you own a manufacturing plant, and you want to know how many employees fall into each employee title category.

In this case, you might configure a query like the one shown in the following image.

In this example, to return the total number of employees with each title, you would need to add the Title field to the query twice. After selecting to show the totals (via the **Query Tasks** menu), you would select **Group By** for the first **Title** field and **Count** for the second **Title** field. You would also want to change the alias of the field whose values you are counting to indicate that it returns a count of employees. Doing so will indicate in the results the total number of employees with each title.

The results might look something like those shown in the following image.

COUNT

Title	# of Employees With Title
Inspector	2
Maintenance Worker	4
Manager	2
Operator	5

From these results, you can see that you have two inspectors, four maintenance workers, two managers, and five operators.

Other Possible Uses

- Total number of failures for specific pieces of equipment.
- Total number of active pieces of equipment in a specific location.
- Total number of pieces of equipment that were installed on or after a specific date.

Available Character Functions

The following table provides a list of the character functions that you can use in queries. The **Oracle Function** column contains the function you will need to use if you are using an Oracle schema. The **SQL Server Function** column contains the function you will need to use if you are using a SQL Server database.

In some cases, there is no SQL Server equivalent of an Oracle function. In these cases, the **SQL Server Function** column contains an empty, shaded cell.

Description	Oracle Function	SQL Server Function
Convert character to ASCII	ASCII	ASCII
Convert ASCII to character	CHR	CHAR
String concatenate	CONCAT	(expression + expression)
Greatest character string in list	GREATEST	
Capitalize first letter of each word in string	INITCAP	
Return starting point of character in character string (from left)	INSTR	CHARINDEX
Least character string in list	LEAST	
Length of character string	LENGTH	DATALENGTH or LEN
Convert characters to lowercase (LOWER)	LOWER	LOWER
Pad left side of character string	LPAD	
Remove leading blanks	LTRIM	LTRIM
Convert string if NULL	NVL	ISNULL
Replace characters	REPLACE	STUFF
String of repeated spaces	RPAD	SPACE
Remove trailing blanks	RTRIM	RTRIM
Combination of letters and numbers that use the Soundex Indexing System to represent the character string*	SOUNDEX	SOUNDEX
Substring	SUBSTR	SUBSTRING

Description	Oracle Function	SQL Server Function
Translate character string	TRANSLATE	
Convert characters to uppercase (UPPER)	UPPER	UPPER

*You can learn more about the Soundex Indexing System by visiting the United States National Archives website.

REPLACE

Suppose you want to view the Failure ID of the failure associated with each piece of equipment or location in your database. Failure IDs are stored with a dash in the syntax, and you want to display them with a double colon instead of the dash.

In this case, you might construct a query like the one shown in the following image.

In this example, to return the failure IDs and replace the dash with a double colon, an expression has been configured using the REPLACE function. The alias has also been configured to indicate that the column displays the Failure ID.

In this case, the expression syntax is:

REPLACE([Failure].[Failure ID], '-', '::')

The syntax indicates that you want to replace the dash (-) with a double colon (::).

The results would look something like those shown in the following image.

REPLACE
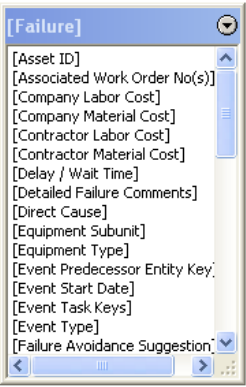
Asset ID	Failure ID
PMP-6010	FAIL::2108
PMP-4499	FAIL::2109
PMP-4917	FAIL::2110
PMP-4543	FAIL::2111
PMP-3871	FAIL::2112
PMP-5382	FAIL::2113
PMP-5382	FAIL::2114
PMP-5382	FAIL::2115
PMP-5382	FAIL::2116

LEN (SQL Only)

Suppose you recently implemented a new process for storing Failure IDs in Failure records. Previously, Failure IDs were formatted as FAIL-n, where n indicated the number of the failure. For example, the tenth failure that was recorded contained a Failure ID of FAIL-10. You have decided that you want all failures to contain four digits and that zeroes should be used as placeholders if the failure is not the 1000th failure or later. For example, the tenth failure should be recorded as FAIL-0010 instead of FAIL-10.

You know that a failure ID in the new format will contain nine characters, where the dash (-) is considered a character. You want to see which Failure records contain IDs with fewer than nine characters.

In this case, you might configure a query like the one shown in the following image.

In this example, to the Failure records with a Failure ID that contains fewer than nine characters, an expression has been configured using the LEN function. The alias has also been configured to indicate that the column displays the number of character in the ID.

In this case, the expression syntax is:

LEN([Failure].[Failure ID])

The query also includes criteria on this column that indicates that the results should display only Failure records where the Failure ID contains *fewer than* nine characters.

A hyperlink has been added to the Failure ID field so that you can open from the results each Failure record in the Record Manager and update the Failure ID. Because the query is running in unformatted mode and the hyperlink includes the EntityKey parameter and the FamilyKey parameter, the Entity Key field and Family Key field are also included in the query.

The results might look something like those shown in the following image.

Failure ID	# of Characters in ID
FAIL-867	8
FAIL-758	8
FAIL-125	8
FAIL-8	6
FAIL-648	8

From these results, you can see that FAIL-867 needs to be reformatted with a Failure ID of FAIL-0867, and so on.

Note: You can use the LEN syntax on a SQL Server database only. To use this function on an Oracle schemas, use LENGTH instead of LEN.

More Examples of Character Functions

The following table lists more examples of using character functions.

Function	Description	Example	Stored Value	Result
CONCAT	Concatenates *two* field values and displays the result.	**CONCAT**([Asset_ID], [Asset_Type])	Asset ID: T-101 Asset Type: Tank	T-101Tank
&	Concatenates *twoor more* field values and adds delimiters between the values.	[Asset_ID] &':' & [Asset_Type]	Asset ID: T-101 Asset Type: Tank	T-101:Tank
LOWER	Displays the value in all lowercase letters.	**LOWER**([Asset_Type])	Tank	tank
UPPER	Displays the value in all capital letters.	**UPPER**([Asset_Type])	Tank	TANK
SUBSTR	Displays a specific number of characters depending on the starting point you specify and the number of characters that you specify should be returned.	**SUBSTR**([Asset_ID],0,3) *Zero (0) specifies the starting point (from left to right) and three (3) specifies the number of characters after the starting point that you want to display.	PMP-101	PMP
NVL	Displays a specified value when a null value is found.	**NVL**([Asset_Type],'No Value Listed')	Asset Type field contains a null value	No Value Listed

Function	Description	Example	Stored Value	Result
INITCAP	Displays the value with the first letter of each word capitalized.	**INITCAP**([Failure_Mode])	bearing failure	Bearing Failure
LTRIM	Displays the value with the specified characters removed from the beginning (left) of the string.	**LTRIM**([PhoneNumber], '(540)-')	(540) 344-9205	344-9205
RTRIM	Displays the value with the specified characters removed from the end (right) of the string.	**RTRIM**([Failure_ID], '-0123456789')	FAIL-1234	FAIL
REPLACE	Displays the value with a specified sequence of characters replaced with another specified set of characters.	**REPLACE**([Asset_ID],'-',':::')	PMP-101	PMP::101

What Are Conversion Functions?

Conversion functions are used to modify query results by reformatting the data. You might want to reformat the data to simplify it (e.g., you could remove unnecessary zeroes from dates, where 01/07/2007 could be converted to 1/7/2007). Or, you might want to reformat the data so that it appears as a different data type completely (e.g., you could spell out a month instead of representing the month with a number, where 01/07/2007 could be converted to January 7, 2007).

Available Conversion Functions

The following table provides a list of the conversion functions that you can use in queries. The **Oracle Function** column contains the function you will need to use if you are using an Oracle schema. The **SQL Server Function** column contains the function you will need to use if you are using a SQL Server database.

Description	Oracle Function	SQL Server Function
Convert a number or date to a character	TO_CHAR	CONVERT
Convert a character to a date	TO_DATE	CONVERT
Convert a character to a number	TO_NUMBER	CONVERT

Example of the TO_CHAR Function (Oracle Only)

Suppose you want to see the only the year in which your shell and tube heat exchangers were installed. All Equipment records in your database contain an Asset Installation Date field, and the values are stored in the format mm/dd/yyyy (e.g., 10/15/2007). You do not want to return the month or day in the query results.

In this case, you might configure a query like the one shown in the following image.
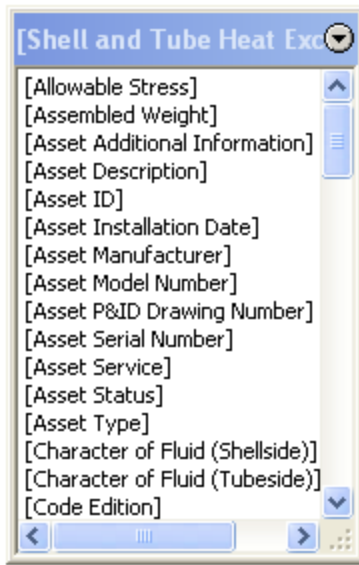
Field	[Asset ID]	[Asset Description]	TO_CHAR([Shell and ...
Alias	Asset ID	Asset Description	Year Installed
Table	[Shell and Tube Heat Exchanger]	[Shell and Tube Heat Exchanger]	
Total	Group By	Group By	Group By
Sort			
Sort Index			
Include	☑	☑	☑
Display	☑	☑	☐
Hyperlink			
Criteria			
Or			

In this example, to return the year in which each shell and tube heat exchanger was installed, an expression has been configured using the TO_CHAR function. The alias has also been configured to indicate that the column returns the year the item was installed.

In this case, the expression syntax is:

TO_CHAR([Shell and Tube Heat Exchanger].[Asset Installation Date], 'yyyy')

The syntax indicates that you want to see the Asset Installation Date field of Shell and Tube Heat Exchanger records and convert the returned values into a yyyy format. In other words, instead of seeing the installation dates in their stored format of mm/dd/yyyy, you will see only the year.

The results might look something like the ones shown in the following image.

Asset ID	Asset Description	Year installed
HE03A	COOLER	2001
HE-360	D1-D4 O.H. TO SLACK WAX MIX	1997
HE-361	P.O. - DI & D4 OVERHEAD	1997
HE-379	D-1 CHARGE PREHEATER	1997
HE-380	D1 & PO PREHEATER	1997
HE-725	SLACK WAX HEATER	1997

If you were to include the Asset Installation Date field in the query *without* converting the returned values, you would see results similar to those shown in the following image.

Asset ID	Asset Description	Asset Installation Date
HE03A	COOLER	3/23/2001
HE-360	D1-D4 O.H. TO SLACK WAX MIX	1/6/1997
HE-361	P.O. - DI & D4 OVERHEAD	1/6/1997
HE-379	D-1 CHARGE PREHEATER	1/6/1997
HE-380	D1 & PO PREHEATER	1/6/1997
HE-725	SLACK WAX HEATER	1/6/1997

Other Possible Uses

- Reformat a numeric cost value so that it represents a currency value (e.g., reformat 125.75 as $125.75).

- Remove the day and year from a date and return only the month (e.g., reformat 12/15/2004 as 12).

The following tables list more examples of using the TO_CHAR function to reformat dates.

Parameter	Description	Example	Stored Value	Result
D	Displays a number representing the day of the week (1-7).	TO_CHAR ([Failure]. [Failure_ Date],'**D**')	12-1-2005	5

Parameter	Description	Example	Stored Value	Result
DAY	Displays the name of the day in all capital letters.	TO_CHAR ([Failure]. [Failure_ Date],' **DAY**')	5-2-2005	MONDAY
Day	Displays the name of the day with the first letter capitalized.	TO_CHAR ([Failure]. [Failure_ Date],' **Day**')	5-2-2005	Monday
DY	Displays the abbreviated name of the day. *This parameter is not case-sensitive. You can type **DY**, **Dy**, **dy**, or **dY** and the result will be displayed with the first letter capitalized.	TO_CHAR ([Failure]. [Failure_ Date],'**DY**')	12-1-2005	Thu
DD	Displays a number rep-resenting the day of the month (1-31).	TO_CHAR ([Failure]. [Failure_ Date],' **DD**')	12-1-2005	01
DDD	Displays a number rep-resenting the day of the year (1-366).	TO_CHAR ([Failure]. [Failure_ Date],' **DDD**')	12-1-2005	335
MM	Displays a number rep-resenting the month of the year (01-12).	TO_CHAR ([Failure]. [Failure_ Date],' **MM**')	12-1-2005	12
MON	Displays the abbreviated name of the month in all capital let-ters.	TO_CHAR ([Failure]. [Failure_ Date],' **MON**')	12-1-2005	DEC

Parameter	Description	Example	Stored Value	Result
Mon	Displays the abbreviated name of the month with the first letter capitalized.	TO_CHAR ([Failure]. [Failure_ Date],' Mon')	12-1-2005	Dec
MONTH	Displays the full name of the month in all capital letters (up to 9 characters).	TO_CHAR ([Failure]. [Failure_ Date],' MONTH')	6-8-2005	JUNE
Month	Displays the full name of the month with the first letter capitalized (up to 9 characters).	TO_CHAR ([Failure]. [Failure_ Date],' Month')	6-8-2005	June
Q	Displays a number representing the quarter of the year (1, 2, 3, 4, where January through March = 1).	TO_CHAR ([Failure]. [Failure_ Date],'Q')	12-1-2005	4
WW	Displays a number representing the week of the year (1-53), where week one starts on the first day of the year and ends on the seventh day of the year.	TO_CHAR ([Failure]. [Failure_ Date],' WW')	12-1-2005	48
W	Displays a number representing the week of the month (1-5), where week one starts on the first day of the month and ends on the seventh day of the month.	TO_CHAR ([Failure]. [Failure_ Date],'W')	12-1-2005	1
YEAR	Displays the name of the year in all capital letters.	TO_CHAR ([Failure]. [Failure_ Date],' YEAR')	12-1-2005 returns	TWO THOUSAND FIVE

Parameter	Description	Example	Stored Value	Result
Year	Displays the name of the year with the first letter capitalized.	TO_CHAR ([Failure]. [Failure_ Date],' Year')	12-1-2005	Two Thou-sand Five
YYYY	Displays a four-digit value rep-resenting the year.	TO_CHAR ([Failure]. [Failure_ Date],' YYYY')	12-1-2005	2005
YYY, YY, Y	Displays a three-, two-, or one-digit value representing the year.	TO_CHAR ([Failure]. [Failure_ Date],'YY')	12-1-2005	05

The following table lists examples of using the TO_CHAR function to reformat numbers.

Parameter	Description	Example	Stored Value	Result
, (comma)	Displays a comma in the specified position.	TO_CHAR([Fail-ure].[Total Fail-ure Cost], '9,999')	1234	1,234
. (period)	Displays a decimal point in the specified position.	TO_CHAR([Fail-ure].[Total Fail-ure Cost], '9,999.00')	1234.11	1,234.11
9	Displays a specific num-ber of digits depending on the number of nines you include. Null values will be displayed for lead-ing zeroes.	TO_CHAR([Fail-ure].[Total Fail-ure Cost], ' 999,999,999 .00')	12345.22	12,345.22
$	Displays a dollar symbol to the left of the value.	TO_CHAR([Fail-ure].[Total Fail-ure Cost], '$ 999,999,999.00')	12345.22	$12,345.22

| Parameter	Description	Example	Stored Value	Result
S	Returns a plus sign (**+**) in the specified position for positive values and a negative sign (-) for negative values.	TO_CHAR([Failure].[Total Failure Cost], '**S** 999,999,999.00')	12345.22	+12,345.22
L	Displays the local currency symbol in this position (as determined by the Windows Regional Settings).	TO_CHAR([Failure].[Total Failure Cost], '**L** 999,999,999.00')	12345.22	$12,345.22 (if local currency is US Dollars)

Available Date Functions

The following table provides a list of the date functions that you can use in query expressions. The **Oracle Function** column contains the function you will need to use if you are using an Oracle schema. The **SQL Server Function** column contains the function you will need to use if you are using a SQL Server database.

In some cases, a function *cannot* be expressed on both an Oracle schema and a SQL Server databases. In these cases, the **Oracle Function** or **SQL Server Function** column contains an empty, shaded cell.

Description	Oracle Function	SQL Server Function
Date addition or subtraction	ADD_ MONTHS MI_DateAdd	DATEADD MI_DateAdd
Last day of month	LAST_DAY	
Difference between dates	MONTHS_ BETWEEN	DATEDIFF
Time zone conversion	NEW_TIME	
Displays the first specified weekday after a stored date	NEXT_DAY	
Substitute a value if it is NULL	NVL	ISNULL
Date round	ROUND	
Re-formats date values	TO_DATE	
Truncates numeric values	TRUNC	
Returns part of a date as a character or numeric value.		DATENAME
Returns part of a date as a numeric value.		DATEPART
Returns part of a date as a numeric value	MI_ DATEPART	MI_ DATEPART
Displays the current date or a date that is earlier or later than the current date	NOW	NOW

ADD_MONTHS (Oracle Only)

Suppose all shell and tube heat exchangers in your facility require inspection every six months. You might want to run a query to see the last inspection date of all shell and tube heat exchangers and the date that is six months after that date. If Shell and Tube Heat Exchanger record in your database contain an *Asset Inspection Date* value, you could do so using the ADD_MONTHS funtion.

You might configure a query like the one shown in the following image.

In this example, to return the next date on which each shell and tube heat exchanger should be inspected, an expression has been configured using the ADD_MONTHS function. The alias has also been configured to indicate that the column returns the next inspection date.

The expression syntax is:

ADD_MONTHS([Shell and Tube Heat Exchanger].[Asset Inspection Date], 6)

The results might look something like those shown in the following image.

Asset ID	Asset Description	Asset Inspection Date	Next Inspection
Exchanger-46-Tubeside		11/1/2009	5/1/2010 12:00:00 AM
Exchanger-5A-Tubeside		9/12/2009	3/12/2010 12:00:00 AM
100101	Chiller Plant Cooling Tower Cell #1	9/8/2009	3/8/2010 1:47:00 PM
100102	Chiller Plant Cooling Tower Cell #2	7/22/2009	1/22/2010 1:47:00 PM
Exchanger-1A-tubeside	REGENERATOR GAS COOLER	7/22/2009	1/22/2010 12:00:00 AM

The following table lists more examples of how you can use the ADD_MONTHS function.
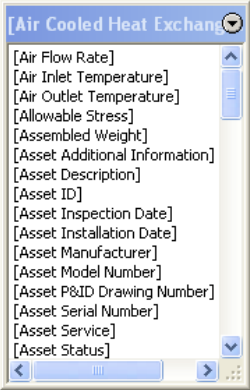
Example Expression	Description	Stored Value	Result
ADD_MONTHS ([Failure].[Failure_ Date],**1**)	Adds one month to the date in the Failure Date field in Failure records and displays the calculated date.	Failure Date: 11/1/2009	12/1/2009
ADD_MONTHS ([Failure].[Failure_ Date],**-1**)	Subtracts one month from the date in the Failure Date field in Failure records and displays the calculated date.	Failure Date: 11/1/2009	10/1/2009

DATEADD (SQL Only)

Suppose all air cooled heat exchangers in your facility require inspection every six months. You might want to run a query to see the last inspection date of all air cooled heat exchangers and the date that is six months after that date. If each Air Cooled Heat Exchanger record in your database contains an *Asset Inspection Date* value, you could do so using the DATEADD function.

You might configure a query like the one shown in the following image.

In this example, to return the last inspection date of each air cooled heat exchanger, the Asset Description field has been added. To return the next inspection date, an expression has been configured using the DATEADD function. The alias of this column has also been configured to indicate that the column returns the next inspection date.

The expression syntax is:

DATEADD('mm', 6, [Air Cooled Heat Exchanger].[Asset Inspection Date])

The results might look something like those shown in the following image.

DATEADD (SQL Only)

Asset ID	Asset Description	Asset Inspection Date ▽	Next inspection Date
HXST 1	SLACK WAX HEATER	10/12/2009	4/12/2010 12:00:00 AM
HXST 100	CHARGE MIX SOLVENT EXCH.	10/6/2009	4/6/2010 12:00:00 AM
HE-2643	DEWAXED OIL\150# STEAM HEATER	7/6/2009	1/6/2010 12:00:00 AM
HXST 10	DEWAXED OIL\150# STEAM HEATER	4/15/2009	10/15/2009 12:00:00 AM
HE-1613	CRUDE BTMS. PUMPAROUND	2/12/2009	8/12/2009 12:00:00 AM
HE-2310	CHARGE MIX SOLVENT EXCH.	1/6/2009	7/6/2009 12:00:00 AM

The following table lists more examples of how you can use the MI_DATEADD function.

Parameter	Example	Description	Stored Value	Result
yy	DATEADD('yy', 1, [Shell and Tube Heat Exchanger].[Asset Inspection Date])	Adds one year to the date in the Asset Inspection Date field in Shell and Tube Heat Exchanger records.	2/12/2009	2/12/2010
mm	DATEADD('mm', 6, [Shell and Tube Heat Exchanger].[Asset Installation Date])	Adds six months to the date in the Asset Installation Date field in Shell and Tube Heat Exchanger records.	2/12/2009	8/12/2009
dd	DATEADD('dd', 4, [Shell and Tube Heat Exchanger].[Asset Installation Date])	Adds four days to the date in the Asset Installation Date field in Shell and Tube Heat Exchanger records.	4/15/2009	4/19/2009
hh	DATEADD('hh', 4, [Shell and Tube Heat Exchanger].[Asset Installation Date])	Adds four hours to the date in the Asset Installation Date field in Shell and Tube Heat Exchanger records.	4/15/2009 4:00:00PM	4/15/2009 8:00:00PM

Parameter	Example	Description	Stored Value	Result
mi	DATEADD('**mi**', **4**, [Shell and Tube Heat Exchanger].[Asset Installation Date])	Adds four minutes to the date in the Asset Installation Date field in Shell and Tube Heat Exchanger records.	4/15/2009 4:00:00PM	4/15/2009 4:04:00PM
ss	DATEADD('**ss**', **4**, [Shell and Tube Heat Exchanger].[Asset Installation Date])	Adds four seconds to the date in the Asset Installation Date field in Shell and Tube Heat Exchanger records.	2/12/2009 1:17:00PM	2/12/2009 1:17:04PM
dw	DATEADD('**dw**', **4**, [Shell and Tube Heat Exchanger].[Asset Installation Date])	Adds four days to the date in the Asset Installation Date field in Shell and Tube Heat Exchanger records.	4/15/2009	4/19/2009
qq	DATEADD('**qq**', **1**, [Shell and Tube Heat Exchanger].[Asset Installation Date])	Adds three months to the date in the Asset Installation Date field in Shell and Tube Heat Exchanger records (i.e., a quarter).	1/6/2009	4/6/2009
dy	DATEADD('**dy**', **2**, [Shell and Tube Heat Exchanger].[Asset Installation Date])	Adds two days to the date in the Asset Installation Date field in Shell and Tube Heat Exchanger records.	1/6/2009	1/8/2009

Parameter	Example	Description	Stored Value	Result
ww	DATEADD('**ww**', **2**, [Shell and Tube Heat Exchanger].[Asset Installation Date])	Adds two weeks to the date in the Asset Installation Date field in Shell and Tube Heat Exchanger records (i.e., 14 days).	1/6/2009	1/20/2009
dd	DATEADD('**dd**',- **4**, [Shell and Tube Heat Exchanger].[Asset Installation Date])	Subtracts four days from the date in the Asset Installation Date field in Shell and Tube Heat Exchanger records.	1/6/2009	1/2/2009

MI_DateAdd

Suppose all air cooled heat exchangers in your facility require inspection every six months. You might want to run a query to see the last inspection date of all air cooled heat exchangers and the date that is six months after that date. If each Air Cooled Heat Exchanger record in your database contains an *Asset Inspection Date* value, you could do so using the MI_DateAdd function.

You might configure a query like the one shown in the following image.

In this example, to return the last inspection date of each air cooled heat exchanger, the Asset Description field has been added. To return the next inspection date, an expression has been configured using the MI_DateAdd function. The alias of this column has also been configured to indicate that the column returns the next inspection date.

The expression syntax is:

MI_DateAdd('mm', 6, [Air Cooled Heat Exchanger].[Asset Inspection Date])

The results might look something like those shown in the following image.

Asset ID	Asset Description	Asset Inspection Date ▽	Next inspection Date
HXST 1	SLACK WAX HEATER	10/12/2009	4/12/2010 12:00:00 AM
HXST 100	CHARGE MIX SOLVENT EXCH.	10/6/2009	4/6/2010 12:00:00 AM
HE-2643	DEWAXED OIL\150# STEAM HEATER	7/6/2009	1/6/2010 12:00:00 AM
HXST 10	DEWAXED OIL\150# STEAM HEATER	4/15/2009	10/15/2009 12:00:00 AM
HE-1613	CRUDE BTMS. PUMPAROUND	2/12/2009	8/12/2009 12:00:00 AM
HE-2310	CHARGE MIX SOLVENT EXCH.	1/6/2009	7/6/2009 12:00:00 AM

The following table lists more examples of how you can use the MI_DATEADD function.

Parameter	Example	Description	Stored Value	Result
yy	MI_DateAdd('**yy**', **1**, [Air Cooled Heat Exchanger].[Asset Installation Date])	Adds one year to the date in the Asset Installation Date field in Air Cooled Heat Exchanger records.	2/12/2009	2/12/2010
mm	MI_DateAdd('**mm**', **6**, [Air Cooled Heat Exchanger].[Asset Installation Date])	Adds six months to the date in the Asset Installation Date field in Air Cooled Heat Exchanger records.	2/12/2009	8/12/2009
dd	MI_DateAdd('**dd**', **4**, [Air Cooled Heat Exchanger].[Asset Installation Date])	Adds four days to the date in the Asset Installation Date field in Air Cooled Heat Exchanger records.	4/15/2009	4/19/2009
hh	MI_DateAdd('**hh**', **4**, [Air Cooled Heat Exchanger].[Asset Installation Date])	Adds four hours to the date in the Asset Installation Date field in Air Cooled Heat Exchanger records.	4/15/2009 4:00:00PM	4/15/2009 8:00:00PM

Parameter	Example	Description	Stored Value	Result
mi	MI_DateAdd('**mi**', **4**, [Air Cooled Heat Exchanger].[Asset Installation Date])	Adds four minutes to the date in the Asset Installation Date field in Air Cooled Heat Exchanger records.	4/15/2009 4:00:00PM	4/15/2009 4:04:00PM
ss	MI_DateAdd('**ss**', **4**, [Air Cooled Heat Exchanger].[Asset Installation Date])	Adds four seconds to the date in the Asset Installation Date field in Air Cooled Heat Exchanger records.	2/12/2009 1:17:00PM	2/12/2009 1:17:04PM
mm	MI_DateAdd('**mi**',**-4**, [Air Cooled Heat Exchanger].[Asset Installation Date])	Subtracts four minutes from the date in the Asset Installation Date field in Air Cooled Heat Exchanger records.	4/15/2009 4:00:00PM	4/15/2009 3:56:00PM

MI_DATEPART

Suppose you want to view only the month in which the shell and tube heat exchangers were installed in your facility. If all Shell and Tube Heat Exchanger records in your database contain an Asset Installation Date field, and the values are stored in the format mm/dd/yyyy (e.g., 11/11/2009), you could do so using the MI_DATEPART function.

You might configure a query like the one shown in the following image.

In this example, to return the month in which each shell and tube heat exchanger was installed, an expression has been configured using the MI_DATEPART function. The alias has also been configured to indicate that the column returns the year the piece of equipment was installed.

The expression syntax is:

MI_DatePart('mm', [Shell and Tube Heat Exchanger].[Asset Installation Date])

Using this syntax, instead of displaying the installation date in the stored format of mm/dd/yyyy, the query results will display only the value representing the month.

The results might look something like those shown in the following image.

Asset ID	Asset Description	Installation Month
▶ HE-07100B	OIL COOLER	3
HE-1288	EFFLUENT FEED-NORTH	1
HE-1289	EFFLUENT FEED-SOUTH	1
HE-1608	CRUDE-HVY. ATMOS. GAS OIL	1
HE-1609	CRUDE-HVY. ATMOS. GAS OIL	1
HE-1612	CRUDE BTMS. PUMPAROUND	1
HE-1613	CRUDE BTMS. PUMPAROUND	1
HE-1666	#1 DEPROP. O.H. COND.	1
HE-1907	UPPER OIL COOLER FOR C-303	1

If you were to include the Asset Installation Date field in the query *without* including the expression to convert the returned values, the results would look something those shown in the following image.

Asset ID	Asset Description	Asset Installation Date
▶ HE-07100B	OIL COOLER	3/23/2001
HE-1288	EFFLUENT FEED-NORTH	1/6/1997
HE-1289	EFFLUENT FEED-SOUTH	1/6/1997
HE-1608	CRUDE-HVY. ATMOS. GAS OIL	1/6/1997
HE-1609	CRUDE-HVY. ATMOS. GAS OIL	1/6/1997
HE-1612	CRUDE BTMS. PUMPAROUND	1/6/1997
HE-1613	CRUDE BTMS. PUMPAROUND	1/6/1997
HE-1666	#1 DEPROP. O.H. COND.	1/6/1997
HE-1907	UPPER OIL COOLER FOR C-303	1/6/1997

The following table lists more examples of how you can use the MI_DATEPART function.

Parameter	Description	Example	Stored Value	Result
yy	Displays only the year from a stored date value.	MI_DatePart('yy', [Shell and Tube Heat Exchanger].[Asset Installation Date])	3/23/2001	2001
mm	Displays only the month from a stored date value.	MI_DatePart('mm', [Shell and Tube Heat Exchanger].[Asset Installation Date])	3/23/2001	3

Parameter	Description	Example	Stored Value	Result
dd	Displays only the day from a stored date value.	MI_DatePart('**dd**', [Shell and Tube Heat Exchanger].[Asset Installation Date])	1/6/1997	6
hh	Displays only the hour from a stored date value.	MI_DatePart('**hh**', [Shell and Tube Heat Exchanger].[Asset Installation Date])	2:05:10PM	14
mi	Displays only the minutes from a stored date value.	MI_DatePart('mi', [Shell and Tube Heat Exchanger].[Asset Installation Date])	2:05:10PM	5
ss	Displays only the seconds from a stored date value.	MI_DatePart('**ss**', [Shell and Tube Heat Exchanger].[Asset Installation Date])	2:05:10PM	10
dw	Displays a number representing the day of the week based on a stored date value.	MI_DatePart('**dw**', [Shell and Tube Heat Exchanger].[Asset Installation Date])	3/23/2001	6
qq	Displays a number representing the quarter of the year based on a stored date value.	MI_DatePart('**qq**', [Shell and Tube Heat Exchanger].[Asset Installation Date])	1/6/2001	1
dy	Displays a number representing the day of the year based on a stored date value.	MI_DatePart('**dy**', [Shell and Tube Heat Exchanger].[Asset Installation Date])	1/6/2001	6

Parameter	Description	Example	Stored Value	Result
ww	Displays a number representing the week of the month based on a stored date value.	MI_DatePart('**ww**', [Shell and Tube Heat Exchanger].[Asset Installation Date])	1/6/2001	2

NOW

Suppose you want to see the tasks that are assigned to a specific user and due in the next thirty days. If Task records contain the *Next Date* field that stores the date on which the task is due, you could do so using the NOW function.

You might configure a query like the one shown in the following image.

In this example, an expression has been configured on the Next Date field using the NOW function. Additionally, a prompt has been configured to prompt the user for the name of the user whose Task records you want to view.

The expression syntax is:

(>= Now() AND <= (Now() + 30))

The syntax indicates that you want to view the tasks that are due today and within the next thirty days. For example, if a task is due thirty-one days after today's date, it will not appear in the query results.

When you run the query, if the current date is 11/18/2009, the results might look something like those shown in the following image.

	Task ID	Next Date	Task Assigned To
▶	Task 1	11/20/2009	User, Meridium
	Task 2	11/25/2009	User, Meridium
	Task 3	11/19/2009	User, Meridium
	Task 4	12/17/2009	User, Meridium

If we were to remove the NOW expression from the query, the results would look something like those shown in the following image, where the first Task record in the image is due thirty-*one* days after the current date.

	Task ID	Next Date	Task Assigned To
▶	Task 5-out of range	12/19/2009	User, Meridium
	Task 1	11/20/2009	User, Meridium
	Task 2	11/25/2009	User, Meridium
	Task 3	11/19/2009	User, Meridium
	Task 4	12/17/2009	User, Meridium

The following table lists more examples of how you can use the NOW function.

Example Expression	Description	Current Date	Result
NOW(**0**) or NOW()	Displays the current date.	11/12/2009	11/12/2009
NOW() **- 3**	Displays the date that is three days prior to the current date.	11/12/2009	11/9/2009
NOW() **+3**	Displays the date that is three days later than the current date.	11/12/2009	11/15/2009

LAST_DAY (Oracle Only)

Suppose you want to see all the air cooled heat exchanger failures in your facility that occurred in a given month. If each Air Cooled Heat Exchanger record is linked to a Failure record, and each Failure record contains the *Failure Date/Time* field that stores the date and time on which a failure occurred, you could do so using the LAST_DAY function.

You might configure a query like the one shown in the following image.

In this example, an expression has been configured using the LAST_DAY function. Additionally, a prompt has been configured that prompts you to select the last day of the month whose failures you want to view. The alias has also been configured to indicate that the column displays the last day of the month.

The expression syntax is:

LAST_DAY([Failure].[Failure Date / Time])

The prompt syntax is:

((? :d :caption='Last Day' :id=Last_Day))

When you run the query, a prompt appears, where you can select 8/31/2009 12:00:00AM to indicate that you want to view failures that occurred during the month of August, as shown in the following image.
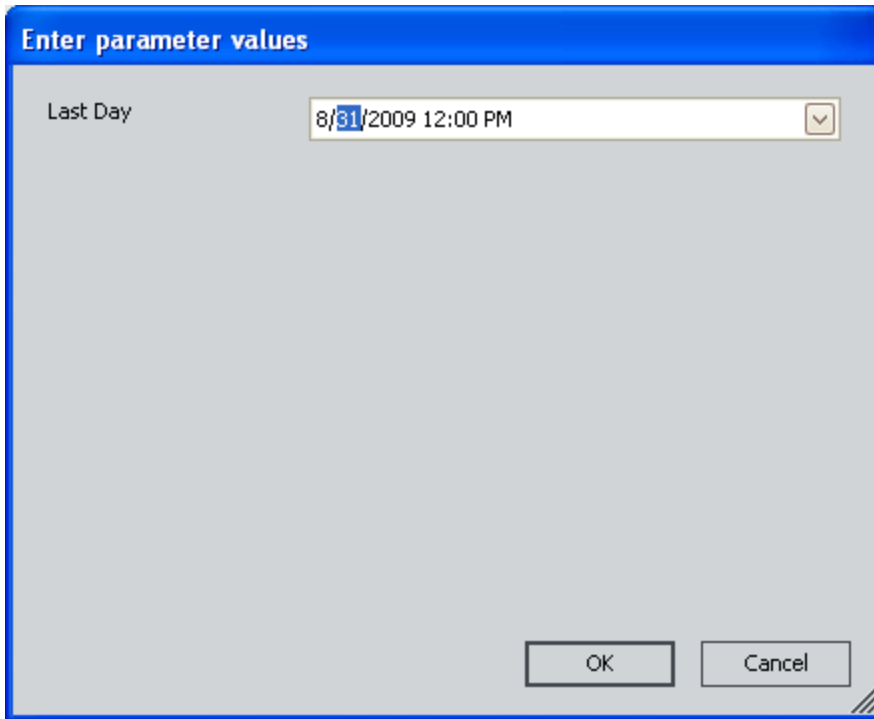
The results might look something like the ones shown in the following image.

Asset ID	Failure ID	Failure Date / Time	Last Day
HX-1799	Failure HX-1799 Kerosene Cooler	8/8/2009	8/31/2009 12:00:00 AM
HX-1806	Failure HX 1806 MID Pump Around Cooler	8/23/2009	8/31/2009 12:00:00 AM
HX-1514	Failure HX 1514 Aire Cooler	8/20/2009	8/31/2009 12:00:00 AM
100101	Failure 100101 Chiller Plant Cooling Tower Cell1	8/3/2009	8/31/2009 12:00:00 AM

NEXT_DAY (Oracle Only)

After pieces of equipment are installed in your facility, you want to perform a quick check on them the following Monday. If all Equipment records in the database contain the field *Installation Date*, you could do so using the NEXT_DAY function.

You might configure a query like the one shown in the following image.
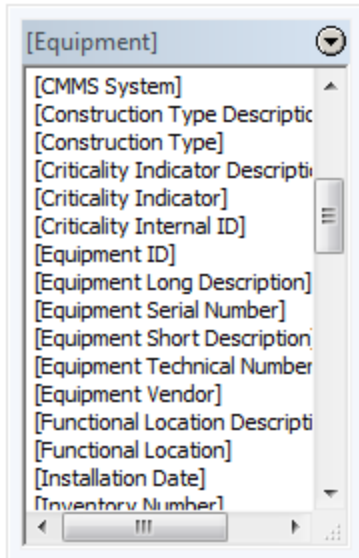
Field	[Equipment ID]	[Equipment Short De...	[Installation Date]	NEXT_DAY([Equipme...
Alias	Equipment ID	Equipment Short Des...	Installation Date	Following Monday
Table	[Equipment]	[Equipment]	[Equipment]	
Total	Group By	Group By	Group By	
Sort				
Sort Index				
Include	✓	✓	✓	✓
Display	✓	✓	☐	✓
Hyperlink				
Criteria	like 'E-%'			
Or				

In this example, an expression has been configured using the NEXT_DAY function. The alias has also been configured to indicate that the column displays the following Monday.

The expression syntax is:

NEXT_DAY([Equipment].[Installation Date], 2)

The NEXT_DAY function assumes that each week begins on Sunday and ends on Saturday. The expression includes a parameter indicating which day of the week you want to see in the results, where the value is between 1 and 7, where 1 represents Sunday and 7 represents Saturday.

In this example, the parameter *2* represents Monday. The syntax indicates that you want to see the date of the first Monday following the date stored in the Installation Date field.

The results might look something like those shown in the following image.

Equipment ID	Equipment Short Description	Following Monday
▶ E-103	Transmitter	1/17/2011 12:00:00 AM
E-101	Indicator	5/29/2000 12:00:00 AM
E-102	Indicator	5/22/2000 12:00:00 AM
E-104	Transmitter	6/26/2006 12:00:00 AM

The following table lists more examples of how you can use the NEXT_DAY function.

Parameters	Example	Description	Stored Value	Result
Sunday Monday Tuesday Wednesday Thursday Friday Saturday	NEXT_DAY([Asset].[Asset Installation Date], '**Sunday**')	Displays the date of the Sunday following the date stored in the Asset Installation Date field.	Tuesday, 11/3/2009	Sunday, 11/8/2009

Parameters	Example	Description	Stored Value	Result
1 2 3 4 5 6 7 **Note:**Each number corresponds to a day of the week in the order in which they occur, where 1 represents Sunday.	NEXT_DAY([Asset].[Asset Installation Date], **1**)	Displays the date of the Sunday following the date stored in the Asset Installation Date field.	Tuesday, 11/3/2009	Sunday, 11/8/2009

MONTHS_BETWEEN (Oracle Only)

After assets have been installed and have failed for the first time, you want to determine how many months passed between the installation date and the failure date. If all Air Cooled Heat Exchanger records contain the field *Asset Installation Date*, and all Failure records contain the field *Failure Date/Time*, you can do so using the MONTHS_BETWEEN function.

You might configure a query like the one shown in the following image.

In this example, an expression has been configured using the MONTHS_BETWEEN function. The alias has also been configured to indicate that the column returns the number of months between installation and failure.

The expression syntax is:

MONTHS_BETWEEN([Failure].[Failure Date / Time], [Air Cooled Heat Exchanger].[Asset Installation Date])

The results might look something like those shown in the following image.

Asset ID	Asset Description	Asset Installation Date	Failure ID	Failure Date / Time	Interval in Months
Exchanger-1A-tubeside	REGENERATOR GAS COOLER	7/22/2009	Failure 1A-Tubeside	11/11/2009	3.64516129032258
HX-1515	AIR COOLER	10/13/2008	Failure HX-1515	10/7/2009	11.8064516129032
HX-1529	AIR COOLER	10/15/2008	Failure HX-1529	10/13/2009	11.9354838709677
100101	Chiller Plant Cooling Tower Cell #1	9/8/2009	Failure 1	11/4/2009	1.87260304659498
100102	Chiller Plant Cooling Tower Cell #2	7/22/2009	Failure of 100102	11/3/2009	3.36857078853047

The values in the **Interval in Months** column contain decimal points. A more usable result would show rounded values without decimal points. You should, therefore, also use the TRUNC function so that the **Interval in Months** column displays only whole numbers.In this case, the entire expression syntax would be:

 TRUNC(MONTHS_BETWEEN([Failure].[Failure Date / Time], [Air Cooled Heat Exchanger].[Asset Installation Date]))

 The results might look something like this:

Asset ID	Asset Description	Asset Installation Date	Failure ID	Failure Date / Time	Interval in Months
Exchanger-1A-tubeside	REGENERATOR GAS COOLER	7/22/2009	Failure 1A-Tubeside	11/11/2009	3
HX-1515	AIR COOLER	10/13/2008	Failure HX-1515	10/7/2009	11
HX-1529	AIR COOLER	10/15/2008	Failure HX-1529	10/13/2009	11
100101	Chiller Plant Cooling Tower Cell #1	9/8/2009	Failure 1	11/4/2009	1
100102	Chiller Plant Cooling Tower Cell #2	7/22/2009	Failure of 100102	11/3/2009	3

DATEDIFF (SQL Only)

After pieces of equipment have been installed and have failed for the first time, you want to determine how many days passed between the installation date and the failure date. If all Air Cooled Heat Exchanger records contain the field *Asset Installation Date*, and all Failure records contain the field *Failure Date/Time*, you can do so using the DATEDIFF function.

You might configure a query like the one shown in the following image.
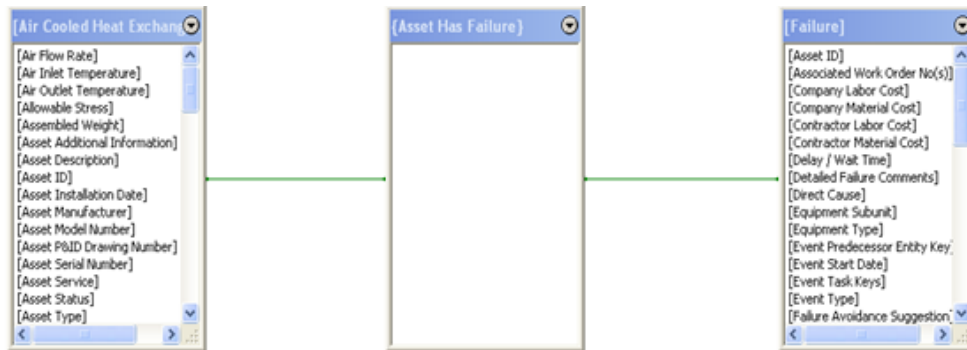
In this example, an expression has been configured using the DATEDIFF function. The alias has also been configured to indicate that the column displays the difference between the installation date and failure date.

In this case, the expression syntax is:

DATEDIFF('day', [Air Cooled Heat Exchanger].[Asset Installation Date], [Failure].[Failure Date / Time])

The results might look something those shown in the following image.

Asset ID	Asset Description	Asset Installation Date	Failure ID	Failure Date / Time	Difference
HX-1810	HVY VAC GAS OIL COOLER	9/8/2009	HX- 1810 Failure	11/4/2009	57
HX-2056	VAC HVY GAS COOLER	11/13/2008	HX-2056 Failure	11/13/2009	365
HX-1811	VAC BTMS COOLER	1/13/2009	HX-1811 Failure	10/31/2009	291
HX-1812	VAC BTMS COOLER	11/13/2000	HX- 1812 Failure	11/2/2009	3276
HX-1813	VAC BTMS COOLER	6/13/2009	HX-1813 Failure	11/12/2009	152

The following table lists more examples of how you can use the DATEDIFF function.

Parameters	Example	Description	Stored Values	Result
yy year	DATEDIFF('**yy**', [Air Cooled Heat Exchanger].[Asset Installation Date], [Failure].[Failure Date / Time])	Displays the difference in years between the dates stored in the Asset Installation Date and Failure Date/Time fields.	Asset Installation Date: 11/12/2008 Failure Date/Time: 11/13/2009	1
mm month	DATEDIFF ('**month**', [Air Cooled Heat Exchanger].[Asset Installation Date], [Failure].[Failure Date / Time])	Displays the difference in months between the dates stored in the Asset Installation Date and Failure Date/Time fields.	Asset Installation Date: 9/8/2009 Failure Date/Time: 11/4/2009	2
dd day dw dy dayofyear	DATEDIFF('**dd**', [Air Cooled Heat Exchanger].[Asset Installation Date], [Failure].[Failure Date / Time])	Displays the difference in days between the dates stored in the Asset Installation Date and Failure Date/Time fields.	Asset Installation Date: 11/13/2008 Failure Date/Time: 11/13/2009	365
hh hour	DATEDIFF('**hour**', [Air Cooled Heat Exchanger].[Asset Installation Date], [Failure].[Failure Date / Time])	Displays the difference in hours between the dates stored in the Asset Installation Date and Failure Date/Time fields.	Asset Installation Date: 9/8/2009 Failure Date/Time: 11/4/2009	1368
mi minute	DATEDIFF('**mi**', [Air Cooled Heat Exchanger].[Asset Installation Date], [Failure].[Failure Date / Time])	Displays the difference in minutes between the dates stored in the Asset Installation Date and Failure Date/Time fields.	Asset Installation Date: 1/13/2009 Failure Date/Time: 10/31/2009	419040

Parameters	Example	Description	Stored Values	Result
ss second	DATEDIFF ('**second**', [Air Cooled Heat Exchanger].[Asset Installation Date], [Failure].[Failure Date / Time])	Displays the difference in minutes between the dates stored in the Asset Installation Date and Failure Date/Time fields.	Asset Installation Date: 6/13/2009 Failure Date/Time: 11/12/2009	13132800
qq quarter	DATEDIFF('**qq**', [Air Cooled Heat Exchanger].[Asset Installation Date], [Failure].[Failure Date / Time])	Displays the difference between the dates stored in the Asset Installation Date and Failure Date/Time fields in increments of quarters.	Asset Installation Date: 9/8/2009 Failure Date/Time: 11/4/2009	1
ww week	DATEDIFF('**week**', [Air Cooled Heat Exchanger].[Asset Installation Date], [Failure].[Failure Date / Time])	Displays the difference in weeks between the dates stored in the Asset Installation Date and Failure Date/Time fields.	Asset Installation Date: 11/13/2008 Failure Date/Time: 11/13/2009	52

TO_DATE (Oracle Only)

Suppose you want to convert the values in the Asset Service field in Air Cooled Heat Exchanger records to dates. For example, the values stored in the Asset Service field look something like the following: 09152008. You want to view this number in date format, like this: 09/15/2008.

In this case, you might configure a query like the one shown in the following image.

In this example, to return the value in the Asset Service field formatted as a date, an expression has been configured using the TO_DATE function. The alias has also been configured to indicate that the column returns the value in the Asset Service field as a date.

In this case, the expression syntax is:

TO_DATE([Air Cooled Heat Exchanger].[Asset Service], 'MMDDYYYY')

...where **'MMDDYYYY'** corresponds to the order in which the values representing the month (MM), day (DD), and year (YYYY) appear in the Asset Service field. Meaning, the values in the Asset Service field appear like this: **09152008**. Where **09** represents the month (MM), **15** represents the day (DD), and **2008** represents the year (YYYY).

The results might look something like those shown in the following image.

Asset ID	Asset Description	Asset Service (Date)
HX-1052	GENERATOR A/C	11/13/2009 12:00:00 AM
Exchanger-5A-Tubeside		7/26/2009 12:00:00 AM
100102	Chiller Plant Cooling Tower Cell #2	4/23/2009 12:00:00 AM
100101	Chiller Plant Cooling Tower Cell #1	1/3/2009 12:00:00 AM
Exchanger-1A-tubeside	REGENERATOR GAS COOLER	9/15/2008 12:00:00 AM
Exchanger-46-Tubeside		5/26/2008 12:00:00 AM
HX-1053	AIR COOLER FOR TURBINE GEN #5	2/4/2005 12:00:00 AM

Note: The format of the parameter shown above ('MMDDYYY') is an example. You should format the parameter to match the values that you want to convert.

NEW_TIME

Suppose times are stored in Air Cooled Heat Exchanger records in Easter Standard Time and you want to convert the values to Pacific Standard Time. If Air Cooled Heat Exchanger records contain the *Asset Installation Date* field, you could do so using the NEW_TIME function.

You might configure a query like the one shown in the following image.

In this example, an expression has been configured using the NEW_TIME function. The alias has also been configured to indicate that the column displays installation dates in Pacific Standard Time.

The expression syntax is:

NEW_TIME([Air Cooled Heat Exchanger].[Asset Installation Date], 'EST', 'PST')

The results might look something like those shown in the following image.

Asset ID	Asset Description	Installation Date (PST)
Exchanger-46-Tubeside	Tubeside exchanger 46	7/22/2009 10:30:02 AM
HX-1529	AIR COOLER	1/16/2009 11:43:17 AM
Exchanger-5A-Tubeside	Tubeside exchanger- 5A	6/16/2008 6:06:03 AM
100102	Chiller Plant Cooling Tower Cell #2	6/2/1997 10:47:00 AM
100101	Chiller Plant Cooling Tower Cell #1	6/2/1997 10:47:00 AM

The following table lists more examples of how you can use the NEW_TIME function.

Parameter	Description	Example	Stored Value	Result
AST ADT	Converts the stored time to Atlantic Standard or Daylight Time.	NEW_TIME([Air Cooled Heat Exchanger].[Asset Installation Date], 'EST', '**AST**')	Asset Installation Date (EST): 6/2/1997 1:47:00PM	Asset Installation Date (AST): 6/2/1997 2:47:00PM
BST BDT	Converts the stored time to Bering Standard or Daylight Time.	NEW_TIME([Air Cooled Heat Exchanger].[Asset Installation Date], 'EST', '**BST**')	Asset Installation Date (EST): 6/2/1997 1:47:00PM	Asset Installation Date BST): 6/2/1997 7:47:00AM
CST CDT	Converts the stored time to Central Standard or Daylight Time.	NEW_TIME([Air Cooled Heat Exchanger].[Asset Installation Date], 'EST', '**CST**')	Asset Installation Date (EST): 6/2/1997 1:47:00PM	Asset Installation Date (CST): 6/2/1997 12:47:00PM
EST EDT	Converts the stored time to Eastern Standard or Daylight Time.	NEW_TIME([Air Cooled Heat Exchanger].[Asset Installation Date], 'PST', '**EST**')	Asset Installation Date (PST): 6/2/1997 10:47:00AM	Asset Installation Date (EST): 6/2/1997 1:47:00PM
GMT	Converts the stored time to Greenwich Mean Time.	NEW_TIME([Air Cooled Heat Exchanger].[Asset Installation Date], 'EST', '**GMT**')	Asset Installation Date (EST): 6/2/1997 1:47:00PM	Asset Installation Date (GMT): 6/2/1997 6:47:00PM
HST HDT	Converts the stored time to Alaska-Hawaii Standard Time or Daylight Time.	NEW_TIME([Air Cooled Heat Exchanger].[Asset Installation Date], 'EST', '**HST**')	Asset Installation Date (EST): 6/2/1997 1:47:00PM	Asset Installation Date (HST): 6/2/1997 8:47:00AM
MST MDT	Converts the stored time to Mountain Standard or Daylight Time.	NEW_TIME([Air Cooled Heat Exchanger].[Asset Installation Date], 'EST', '**MST**')	Asset Installation Date (EST): 6/2/1997 1:47:00PM	Asset Installation Date (MST): 6/2/1997 11:47:00AM

Parameter	Description	Example	Stored Value	Result
NST	Converts the stored time to Newfoundland Standard Time.	NEW_TIME([Air Cooled Heat Exchanger].[Asset Installation Date], 'EST', '**NST**')	Asset Installation Date (EST): 6/2/1997 1:47:00PM	Asset Installation Date (NST): 6/2/1997 3:17:00PM
PST PDT	Converts the stored time to Pacific Standard or Daylight Time.	NEW_TIME([Air Cooled Heat Exchanger].[Asset Installation Date], 'EST', '**PST**')	Asset Installation Date (EST): 6/2/1997 1:47:00PM	Asset Installation Date (PST): 6/2/1997 10:47:00AM
YST YDT	Converts the stored time to Yukon Standard or Daylight Time.	NEW_TIME([Air Cooled Heat Exchanger].[Asset Installation Date], 'EST', '**YST**')	Asset Installation Date (EST): 6/2/1997 1:47:00PM	Asset Installation Date (YST): 6/2/1997 9:47:00AM

NVL (Oracle Only)

Suppose you want to see the serial number of the air cooled heat exchangers in your facility even if the Asset Serial Number field in the Air Cooled Heat Exchanger records is empty, and if the Asset Serial Number field is empty, you want to see the value *Not Defined*.

In this case, you might configure a query like the one shown in the following image.

In this example, an expression has been configured using the NVL function. The alias has also been configured to indicate that the column returns serial numbers.

In this case, the expression syntax is:

NVL([Air Cooled Heat Exchanger].[Asset Serial Number], 'Not Defined')

...where **Not Defined** is the value that will appear where a null value is found in the Asset Serial Number field in the Air Cooled Heat Exchanger records.

The results might look something like those shown in the following image.

NVL (Oracle Only)

Asset ID	Asset Description	Serial Numbers (ALL)
100101	Chiller Plant Cooling Tower Cell #1	NC902131
100102	Chiller Plant Cooling Tower Cell #2	NC902131
HX-2521	TEMPERED WATER COOLER	Not Defined
HX-2522	TEMPERED WATER COOLER	Not Defined

ISNULL (SQL Only)

Suppose you want to see the installation date of the shell and tube heat exchangers in your facility even if the Asset Installation Date field in the Shell and Tube Heat Exchanger records is empty, and if the Asset Installation Date field is empty, you want to see the value *1/01/2001*.

In this case, you might configure a query like the one shown in the following image.

In this example, an expression has been configured using the ISNULL function. The alias has also been configured to indicate that the column returns the installation dates.

In this case, the expression syntax is:

IsNull([Shell and Tube Heat Exchanger].[Asset Installation Date], '1/01/2001')

...where **1/01/2001** is the value that will appear where a null value is found in the Asset Installation Date field in the Shell and Tube Heat Exchanger records.

The results might look something like those shown in the following image.
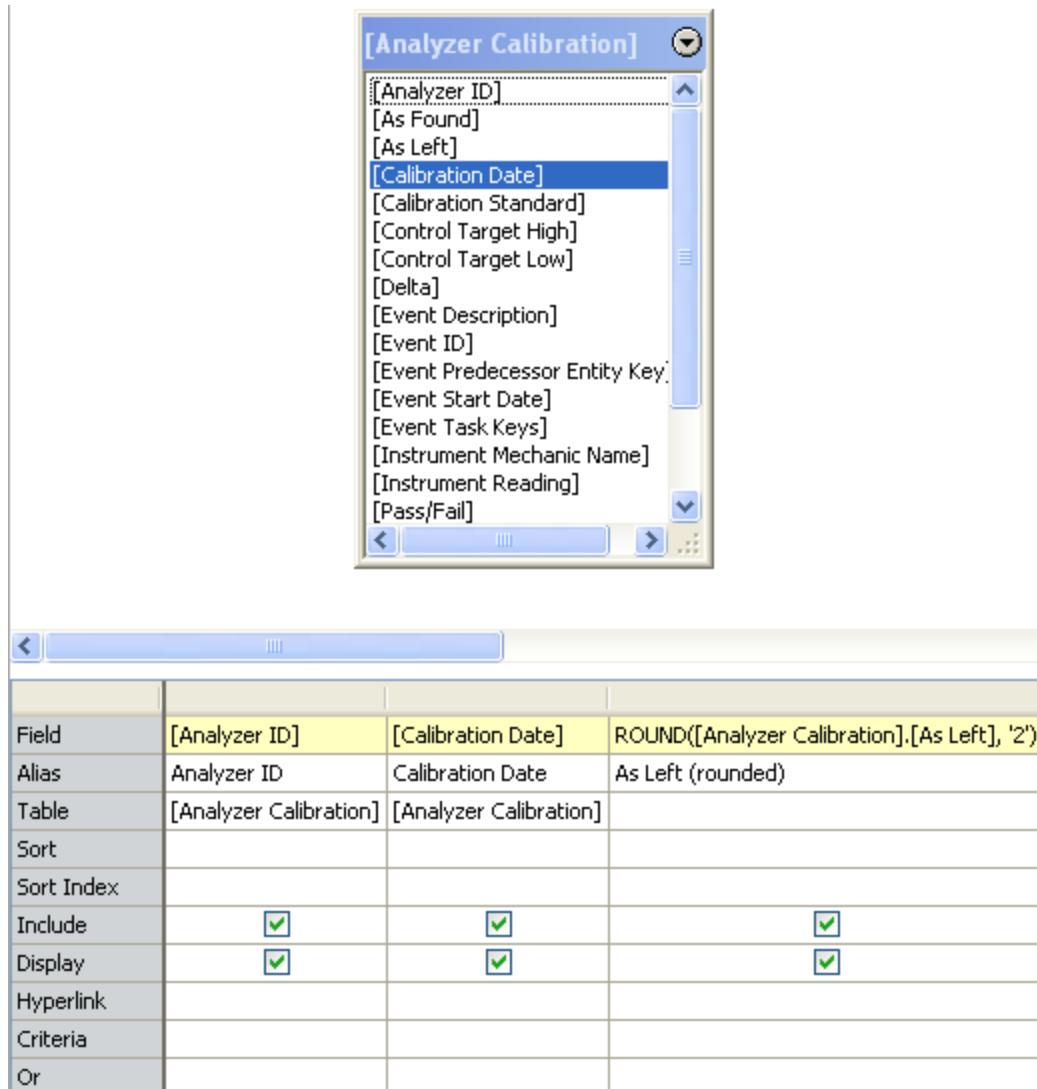
Asset ID	Asset Description	Installation Dates (ALL)
HE03A	COOLER	3/23/2001 12:00:00 AM
HE-07100B	OIL COOLER	3/23/2001 12:00:00 AM
HXST 99	REACTOR EFFLUENT COOLER	1/1/2001 12:00:00 AM
HXST 98	#4 SULFUR CONDENSER	1/1/2001 12:00:00 AM
HXST 97	NO. 3 SULFUR CONDENSER	1/1/2001 12:00:00 AM
HXST 96	NO. 2 REHEATER	1/1/2001 12:00:00 AM

ROUND (Oracle Only)

Suppose you want to see the values in the *As Left* fields for the analyzer instruments in your facility rounded to two decimal places. If all Analyzer Calibration records contain the As Left field, you can do so using the ROUND function.

You might configure a query like the one shown in the following image.

In this example, an expression has been configured using the ROUND function. The alias has also been configured to indicate that the column displays the rounded *as left* values.

The expression syntax is:

ROUND([Analyzer Calibration].[As Left], '2')

...where **2** is the number of decimal places to which the values will be rounded in the results.

The results might look something like those shown in the following image.

Analyzer ID	Calibration Date	As Left (rounded)	▽
10542	1/11/2000		10.59
10542	1/11/2000		10.49
10542	1/11/2000		10.49
10542	1/11/2000		10.32
10260	1/11/2000		10.04

TRUNC (Oracle Only)

Suppose you want to see the number of months between when the air cooled heat exchangers were installed and when they failed for the first time, and you want to view that number in a truncated form. For example, instead of seeing the number of months in the format *3.33333*, you want to see only *3*. If all Air Cooled Heat Exchanger records contain the field *Asset Installation Date*, and all Failure records contain the field *Failure Date/Time*, you could do so using the TRUNC function.

You might configure a query like the one shown in the following image.

In this example, an expression has been configured using the TRUNC function and the MONTHS_BETWEEN function. The alias has also been configured to indicate that the column returns the interval between installation and failure.

The expression syntax is:

TRUNC(MONTHS_BETWEEN([Failure].[Failure Date / Time], [Air Cooled Heat Exchanger].[Asset Installation Date]))

The results might look something like those shown in the following image.

Asset ID	Asset Description	Asset Installation Date	Failure ID	Failure Date / Time	Interval in Months
Exchanger-1A-tubeside	REGENERATOR GAS COOLER	7/22/2009	Failure 1A-Tubeside	11/11/2009	3
HX-1515	AIR COOLER	10/13/2008	Failure HX-1515	10/7/2009	11
HX-1529	AIR COOLER	10/15/2008	Failure HX-1529	10/13/2009	11
100101	Chiller Plant Cooling Tower Cell #1	9/8/2009	Failure 1	11/4/2009	1
100102	Chiller Plant Cooling Tower Cell #2	7/22/2009	Failure of 100102	11/3/2009	3

DATENAME

Suppose you want to see the name of the month in which the shell and tube heat exchangers were installed. Instead of *11*, you want to see *November*, and you do not want to return the year or day in the query results. If all Shell and Tube Heat Exchanger records in your database contain an *Asset Installation Date* field, and the values are stored in the format mm/dd/yyyy (e.g., 11/11/2009), you could do so using the DATENAME function.

You might configure a query like the one shown in the following image.
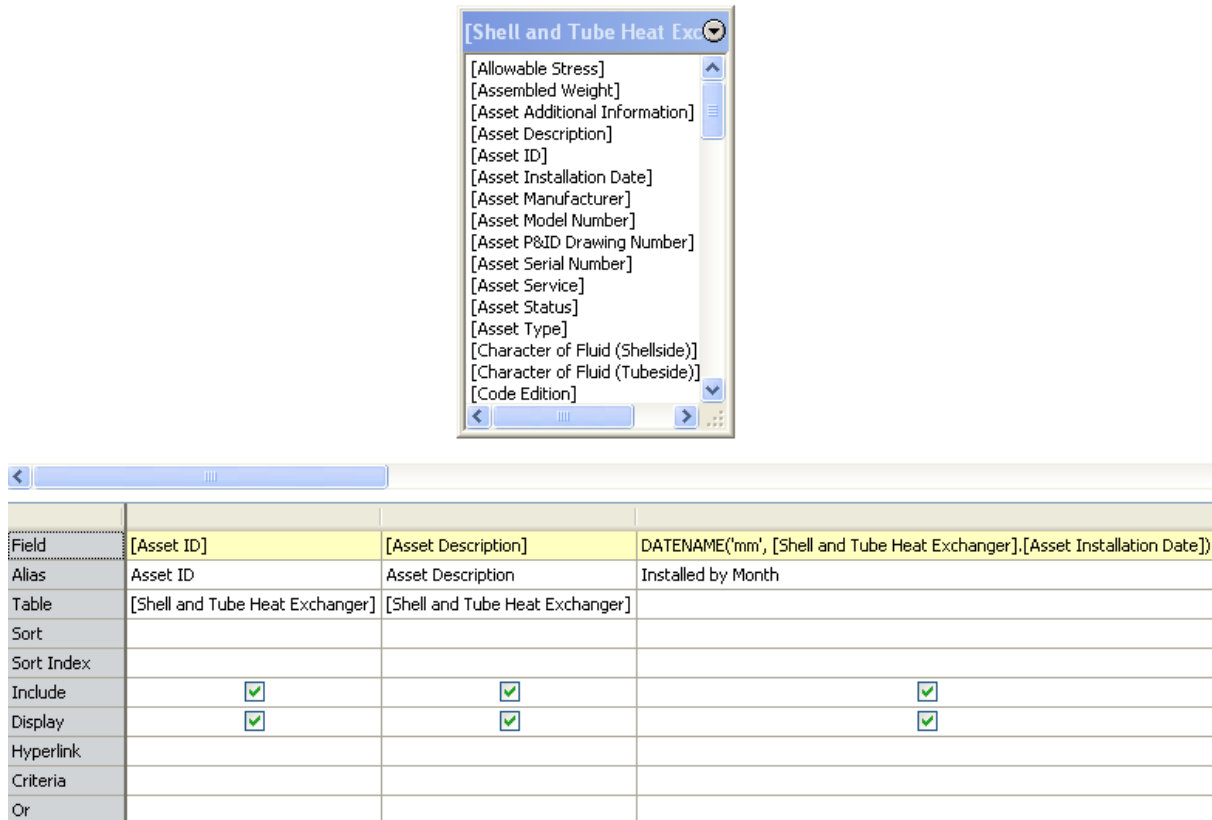
In this example, an expression has been configured using the DATENAME function. The alias has also been configured to indicate that the results display the month.

The expression syntax is:

DATENAME('mm', [Shell and Tube Heat Exchanger].[Asset Installation Date])

...where **mm** indicates the part of the date (i.e., month) that you want to convert to its character format.

The results might look something like those shown in the following image.

Asset ID	Asset Description	Installed by Month
HE-2943	LEAN/RICH AMINE EXCHANGER	January
HE-2944	LEAN/RICH AMINE EXCHANGER	January
HE-2945	LEAN/RICH AMINE EXCHANGER	January
HE-2950	NO.1 STEAM REHEATER	January
HE-2554	AIR AFTERCOOLER (C-454)	April
HE-07100B	OIL COOLER	March
HE03A	COOLER	March

The following table lists more examples of how you can use the DATENAME function.

Parameter	Description	Example	Stored Value	Result
yy year	Displays a number representing the year from a stored date value.	DATENAME('**year**', [Shell and Tube Heat Exchanger].[Asset Installation Date])	1/6/1997	1997
mm month	Displays the month from a stored date value.	DATENAME('**mm**', [Shell and Tube Heat Exchanger].[Asset Installation Date])	2/12/2001	February
dd day	Displays a number representing the day from a stored date value.	DATENAME('**day**', [Shell and Tube Heat Exchanger].[Asset Installation Date])	1/23/1997	23
dw	Displays the day of the week based on a stored date value.	DATENAME('**dw**', [Shell and Tube Heat Exchanger].[Asset Installation Date])	1/5/1997	Monday
qq quarter	Displays a number representing the quarter of the year based on a stored date value.	DATENAME('**qq**', [Shell and Tube Heat Exchanger].[Asset Installation Date])	3/23/2001	1
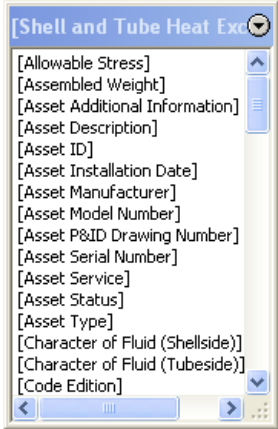
Parameter	Description	Example	Stored Value	Result
dy dayofyear	Displays a number rep-resenting the day of the year based on a stored date value.	DATENAME('**dayofyear**', [Shell and Tube Heat Exchanger].[Asset Install-ation Date])	1/23/2001	23
ww	Displays a number rep-resenting the week of the month based on a stored date value.	DATENAME('**ww**', [Shell and Tube Heat Exchanger].[Asset Installation Date])	1/6/1997	2

DATEPART

Suppose you want to see the year in which the shell and tube heat exchangers were installed in your facility. If all Shell and Tube Heat Exchanger records in your database contain an *Asset Installation Date* field, and the values are stored in the format mm/d-d/yyyy (e.g., 11/11/2009), you could do so using the DATEPART function.

You might configure a query like the one shown in the following image.

In this example, an expression has been configured using the DATEPART function. The alias has also been configured to indicate that the column displays the installation year.

The expression syntax is:

DATEPART('year', [Shell and Tube Heat Exchanger].[Asset Installation Date])

...where **year** is the portion of the date value that you want to view.

The results might look something like those shown in the following image.

The following table lists more examples of how you can use the DATEPART function.

Parameter	Description	Example	Stored Value	Result
yy year	Displays a number representing the year from a stored date value.	DATEPART('**yy**', [Shell and Tube Heat Exchanger]. [Asset Installation Date])	1/6/1997	1997
mm month	Displays a number representing the month from a stored date value.	DATEPART('**mm**', [Shell and Tube Heat Exchanger]. [Asset Installation Date])	2/12/2001	2
dd day	Displays a number representing the day from a stored date value.	DATEPART('**dd**', [Shell and Tube Heat Exchanger]. [Asset Installation Date])	1/23/1997	23
dw	Displays a number representing the day of the week based on a stored date value.	DATEPART('**dw**', [Shell and Tube Heat Exchanger]. [Asset Installation Date])	1/6/1997	2 (i.e., Monday)
qq quarter	Displays a number representing the quarter of the year based on a stored date value.	DATEPART('**quarter**', [Shell and Tube Heat Exchanger]. [Asset Installation Date])	1/23/2001	1
dy dayofyear	Displays a number representing the day of the year based on a stored date value.	DATEPART('**dy**', [Shell and Tube Heat Exchanger]. [Asset Installation Date])	1/23/2001	23

Parameter	Description	Example	Stored Value	Result
ww week	Displays a number rep-resenting the week of the month based on a stored date value.	DATEPART('**week**', [Shell and Tube Heat Exchanger].[Asset Installation Date])	1/6/1997	2

Available Number/Mathematical Functions

The following table provides a list of the number/mathematical functions that you can use in queries. The **Oracle Function** column contains the function you will need to use if you are using an Oracle schema. The **SQL Server Function** column contains the function you will need to use if you are using a SQL Server database.

In some cases, a function cannot be expressed on both an Oracle schema and a SQL Server databases. In these cases, the **Oracle Function** or **SQL Server Function** column contains an empty, shaded cell.

Description	Oracle Function	SQL Server Function
Absolute value	ABS	ABS
Arc cosine	ACOS	ACOS
Arc sine	ASIN	ASIN
Arc tangent of n	ATAN	ATAN
Arc tangent of n and m	$ATAN_2$	$ATAN_2$
Smallest integer>=value	CEIL	CEILING
Cosine	COS	COS
Hyperbolic cosine	COSH	COT
Exponential value	EXP	EXP
Largest integer<=value	FLOOR	FLOOR
Natural logarithm	LN	LOG
Logarithm, any base	LOG(N)	LOG(N)
Logarithm, base 10	LOG(10)	LOG_{10}
Modulus (remainder)	MOD	USE MODULO(%) OPERATOR
Power	POWER	POWER
Random number		RAND
Round	ROUND	ROUND
Sign of number	SIGN	SIGN
Sine	SIN	SIN

Description	Oracle Function	SQL Server Function
Hyperbolic sine	SINH	
Square root	SQRT	SQRT
Tangent	TAN	TAN
Hyperbolic tangent	TANH	
Truncate	TRUNC	
Largest number in list	GREATEST	
Smallest number in list	LEAST	
Convert number if NULL	NVL	ISNULL

About the DECODE Function

In addition to the aggregate, character, conversion, date, and number/mathematical functions, you can also use the DECODE function, which evaluates a value in the query results and based on that evaluation, triggers an action or returns a different value. *The DECODE function can be used for Oracle schemas only.*

Decode Example

Suppose that in your organization, you have open and closed work history events. Each Work History record contains an Order System Status field, which is used to record the status of that work history event. You want to see how many open work history events you have and how many closed work history events you have. Instead of seeing the stored values of *CLSD TECO* or *OPEN*, however, you want to see the values *Closed* or *Open* in your query results.

In addition, you are concerned that not all Work History records contain a value in the Order System Status field. You also want to see from your query results which records do not have a value in this field so that you can update those records.

In this case, you might configure a query like the one shown in the following image.

Field	Decode([Work History].[Order System Status],...	[Order System Status]
Alias	Expr	Order System Status
Table		[Work History]
Total	Group By	Count
Sort		
Sort Index		
Include	☑	☑
Display	☑	☑
Hyperlink		
Criteria		
Or		

In this example, to return the status of each work history event as Closed, Open, or No Status (meaning that the record does not contain a value in the Order System Status field), an expression has been configured using the DECODE function.

In this case, the expression syntax is:

Decode([Work History].[Order System Status], 'CLSD TECO', 'Closed', 'OPEN', 'Open', 'No Status')

This syntax indicates that for Work History records that contain the value *CLSD TECO* in the Order System Status field, you want to return the value *Closed*. For those records with the value *OPEN* in the Order System Status field, you want to return the value *Open*. For records where the Order System Status field is empty, you want to return the value **No Status**.

This query also uses a COUNT function on the Order System Status field so that you can see the number of work history events that fall into each category.

The results might look something like those shown in the following image.

Status	Number
Closed	3
No Status	0
Open	1

From these results, you can see that you have three closed work history events, zero work history events with no status, and one open work history event.

Other Possible Uses of the DECODE Function

You might want to use the DECODE function in the following situations (or similar ones):

- You want to review the types of maintenance activities that are being performed in your plant and the total number of failures that resulted in each type of activity. Users can type any value they choose in a field that tracks maintenance activities, so you know that your users are using different terminology to mean the same thing. For example, some users type **replace**, some users type **replacing**, and others type **replacement** to indicate that they replaced a broken component. You want to write a decode statement to indicate that the values **replace**, **replacing**, and **replacement** should yield the value **Replace** in the query results.

- Members of management want to investigate failures that resulted in a failure cost of $50,000 or more. You decide to write a decode statement to indicate that failures with a failure cost greater than or equal to $50,000 should yield the result **Please Investigate** in the query results. Other failures should yield the result **OK** in the query results.

What is a WHERE Clause?

A WHERE clause defines conditions that you want to apply to a query. In a non-aggregate query, WHERE clauses are used instead of HAVING clauses. In an [aggregate query](), a WHERE clause defines conditions that you want to apply before any calculations are performed.

For instance, you might want to calculate the total failure cost for all pieces of equipment, but you only want to include in the calculation failures with a total failure cost greater than $5,000.00. You could create a query like this:

SELECT [Asset].[ASSET_ID_CHR] "Asset ID", Sum([Failure].[EFAIL_TOTCST_FRM]) "Total Failure Cost"
FROM [Asset] JOIN_SUCC [Failure] ON {Asset Has Failure}
WHERE [Failure].[EFAIL_TOTCST_FRM] > 5000
GROUP BY [Asset].[ASSET_ID_CHR]

In this query, you can see that the WHERE clause is:

WHERE [Failure].[EFAIL_TOTCST_FRM] > 5000

This WHERE clause returns the total failure cost for all pieces of equipment but only calculates the failures with a total failure cost greater than $5,000.00.

In other words, in an aggregate query, WHERE clauses are used to define conditions that you want to apply to the query *before* the calculation dictated by the aggregate function has been performed. In the previous example, a piece of equipment might have failures whose failure costs were $5,050.00, $1,000.00, and $500.00. Because the WHERE clause indicates that you want to return the total failure cost for all pieces of equipment but only calculate the failures with a total failure cost greater than $5,000.00, the total failure cost returned for this piece of equipment would be $5,050.00 (the failures with a cost of $1,000.00 and $500.00 are not included in the calculation because they are less than $5,000.00 each).

How Do I Create a WHERE Clause?

You can create a WHERE clause by:

- Typing it manually in the SQL code.

- Typing anything in the **Criteria** cell or the **Or** cell when the design grid does not contain the **Total** cell.

Field	[Asset ID]	[Total Failure Cost]
Alias	Asset ID	Total Failure Cost
Table	[Asset]	[Failure]
Sort		
Sort Index		
Include	☑	☑
Display	☑	☑
Hyperlink		
Criteria		(> 5000) ⬅
Or		

- Typing anything in the **Criteria** cell or the **Or** cell when the same column contains the text **Where** in the **Total** cell.

Field	[Asset ID]	[Total Failure Cost]
Alias	Asset ID	Total Failure Cost
Table	[Asset]	[Failure]
Total	Group By	Where ⬅
Sort		
Sort Index		
Include	☑	☐
Display	☑	☐
Hyperlink		
Criteria		(> 5000) ⬅
Or		

> Œ**Hint:** If the **Total** cell contains the text **Where**, make sure to *clear* the **Include** and **Display** check boxes in that column.

Note that the text **WHERE** will not appear in the **Criteria** cell or the **Or** cell. It will, however, appear in the SQL code.

What is a HAVING Clause?

A HAVING clause defines conditions that you want to apply to an aggregate query. After the aggregate function has been executed, the condition you define is applied to each aggregate value (i.e., the value that is calculated as a result of the aggregate function).

For instance, you might want to see all pieces of equipment whose failures resulted in a total failure cost greater than $5,000.00. You could create a query like this:

SELECT [Asset].[ASSET_ID_CHR] "Asset ID", Sum([Failure].[EFAIL_TOTCST_FRM]) "Total Failure Cost", Count([Failure].[MI_EVENT_ID]) "Failure Count"
FROM [Asset] JOIN_SUCC [Failure] ON {Asset Has Failure}
GROUP BY [Asset].[ASSET_ID_CHR]
HAVING Sum([Failure].[EFAIL_TOTCST_FRM]) > 5000

In this query, you can see that the HAVING clause is:

HAVING Sum([Failure].[EFAIL_TOTCST_FRM]) > 5000

This HAVING clause returns pieces of equipment whose failures resulted in a total failure cost greater than $5,000.00.

In other words, in an aggregate query, HAVING clauses are used to define conditions that you want to apply to the query *after* the calculation dictated by the aggregate function has been performed. In the previous example, a piece of equipment might have failures whose failure costs were $3000.00, $1,000.00, and $500.00. If you add these values together, you can see that the total failure cost for failures associated with the piece of equipment is $4,500.00. Because the HAVING clause indicates that you want to return only pieces of equipment with failures with a *total* failure cost greater than $5,000.00, this piece of equipment would not be returned ($4,500.00 is less than $5,000.00).

How Do I Create a HAVING Clause?

You can create a HAVING clause by:

- Typing it manually in the SQL code.

- Typing anything in the **Criteria** cell or the **Or** cell when the same column contains anything *other than* **Where** in the **Total** cell.

Note that the text **HAVING** will not appear in the **Criteria** cell or the **Or** cell. It will, however, appear in the SQL code.

What is an Operator?

An operator is a symbol or combination of symbols that indicates how you want to use a field value in an expression. For example, you might use the greater than (>) operator to indicate that you want to return records where the field value is greater than a specified number (e.g., >100).

Valid operators vary for different field types. This documentation explains the operators you should use with the following fields types:

- Character fields
- Text fields
- Date fields
- Logical fields
- Numeric fields

Operators to Use with Character Fields

The following table provides examples of operators that you can use to manipulate values returned in character fields.

Note that when you type anything in the **Criteria** cell or the **Or** cell, the *is equal to* (=) operator is assumed *at the beginning* of the text unless a different operator is explicitly specified. For example, typing simply `Rotating Pump` in the **Criteria** cell of a numeric field causes the Meridium APM system to assume that you want to return records where the value in that field *is equal to Rotating Pump*.

You can type the **=** operator at the beginning of the text, but it is not necessary. If you do not type it, the Meridium APM system will insert it automatically, but it will not be displayed. In addition, the Meridium APM system inserts but does not display the text WHERE or HAVING before the beginning operator (either an implied = operator or an explicit alternative operator). This means that the SQL code will contain either a WHERE clause or a HAVING clause as a result of the text that you type in the **Criteria** cell or the **Or** cell.

Purpose	Operator	Example	Outcome of Example
Return records that have an exact value stored in a given field.	=	='Rotating Pump'	Returns records where the field contains the exact value **Rotating Pump**.*
Return records that have one value or another stored in a single field.	Or	='Rotating Pump' or 'Centrifugal Pump'	Returns records where the value stored in the field is **Rotating Pump** and records where the value stored in the field is **Centrifugal Pump**.
Return records that do not have a specified value in a given field.	Not != <>	Not 'Centrifugal Pump' != 'Centrifugal Pump' <> 'Centrifugal Pump'	Returns records where the field does not contain the value **Centrifugal Pump**.

Purpose	Operator	Example	Outcome of Example
Return records where a given field contains the specified value.	Like	Like '%Pump'	Returns records where the specified field contains the term **Pump**, preceded by any number of characters, such as Centrifugal (Centrifugal Pump) or Rotating (Rotating Pump).
Return records where a given field contains the specified value.	Like	Like 'Pump_'	Returns records where the specified field contains the term **Pump**, followed by any one character, such as 1 (Pump1).
Return records where a given field does not contain the specified value.	Not Like	Not Like 'Centrifugal%'	Returns records where the field value does not contain **Centrifugal** followed by any number of characters. For example, this would eliminate Centrifugal Pump, Centrifugal Pump A, and Centrifugal Pump1.
Return records where a given field does not contain the specified value.	Not Like	Not Like 'Pump_'	Returns records where the field value does not contain **Pump** followed by any one character. For example, this would eliminate PumpA and Pump1.

Purpose	Operator	Example	Outcome of Example
Return records that have multiple, specified values stored in a given field.*	|(Pipe)	Like 'Bob|John'	Returns records where the field contains both the value **Bob** and the value **John**.
Return records that contain any values in a given field.	Is Not Null	Is Not Null	Returns records where the field contains any value.
Returns records that do not contain any values in a given field.	Is Null	Is Null	Returns records where the field is empty.

* Use *like* instead of **=** in [multi-value character fields](). Using = will return an error. Using *like*, however, will return the same results that you would expect to get using the = operator. In multi-value character fields, use the pipe character (\|) to separate multiple values.

Operators to Use with Text Fields

The following table provides examples of operators that you can use to manipulate values returned in text fields. Note that multi-value character fields are stored like text fields; the values are stored as a continuous string, where each value is separated by the | (pipe) character. Therefore, instead of using the operators allowed for character fields, when querying on a multi-value character fields, use the operators described in the table.

You can type the **=** operator at the beginning of the text, but it is not necessary. If you do not type it, the Meridium APM system will insert it automatically, but it will not be displayed. In addition, the Meridium APM system inserts but does not display the text WHERE or HAVING before the beginning operator (either an implied = operator or an explicit alternative operator). This means that the SQL code will contain either a WHERE clause or a HAVING clause as a result of the text that you type in the **Criteria** cell or the **Or** cell.

Purpose	Operator	Example Expression	Outcome of Example
Return records where a given field contains the specified value.	Like	Like '%Pump'	Returns records where the field contains the term **Pump**, preceded by any number of characters, such as Centrifugal (Centrifugal Pump) or Rotating (Rotating Pump).
Return records where a given field contains the specified value.	Like	Like 'Pump_'	Returns records where the field contains the term **Pump**, followed by any one character, such as 1 (Pump1).

Purpose	Operator	Example Expression	Outcome of Example
Return records where a given field does not contain the specified value.	Not Like	Not Like 'Centrifugal%'	Returns records where the field value does not contain **Centrifugal** followed by any number of characters. For example, this would eliminate Centrifugal Pump, Centrifugal Pump A, and Centrifugal Pump1.
Return records where a given field does not contain the specified value.	Not Like	Not Like 'Pump_'	Returns records where the field value does not contain **Pump** followed by any one character. For example, this would eliminate PumpA and Pump1.
Return records that have multiple, specified values stored in a given field.	|(Pipe)	Like 'Bob|John'	Returns records where the field contains both the value **Bob** and the value **John**. Note that this example applies only to multi-value character fields.
Return records that contain any values in a given field.	Is Not Null	Is Not Null	Returns records where the field contains any value.
Returns records that do not contain any values in a given field.	Is Null	Is Null	Returns records where the field is empty.

Operators to Use with Date Fields

The following table provides examples of operators that you can use to manipulate values returned in date fields.

Note that when you type anything in the **Criteria** cell or the **Or** cell, the *is equal to* (=) operator is assumed *at the beginning* of the text unless a different operator is explicitly specified. For example, typing simply `(#:D '2000-03-10')` in the **Criteria** cell of a date field causes the Meridium APM system to assume that you want to return records where the value in that field *is equal to March 10, 2000*.

You can type the **=** operator at the beginning of the text, but it is not necessary. If you do not type it, the Meridium APM system will insert it automatically, but it will not be displayed. In addition, the Meridium APM system inserts but does not display the text WHERE or HAVING before the beginning operator (either an implied = operator or an explicit alternative operator). This means that the SQL code will contain either a WHERE clause or a HAVING clause as a result of the text that you type in the **Criteria** cell or the **Or** cell.

Purpose	Operator	Example	Outcome of Example
Return records that have an exact date stored in a field.	=	=(#:D '2000-03-10')	Returns records where the date is March 10, 2000 and the time is any time.
Return records that have an exact date and time stored in a field.	=	=(# :dt '2006-01-01 17:00:00')	Returns records where the date is January 1, 2006 and the time is 5:00 PM.
Return records that have a date greater than the specified date in a given field.	>	>(#:D '2000-03-10')	Returns records where the date is after March 10, 2000.
Return records that have a date less than the specified date in a given field.	<	<(#:D '2000-03-10')	Returns records where the date is before March 10, 2000.

Purpose	Operator	Example	Outcome of Example
Return records that have a date greater than or equal to the specified date in a given field.	>=	>=(#:D '2000-03-10')	Returns records where the date is on or after March 10, 2000.
Return records that have a date less than or equal to the specified date in a given field.	<=	<=(#:D '2000-03-10')	Returns records where the date is on or before March 10, 2000.
Return records that do not have the specified date in a given field.	Not !=	Not (#:D '2000-03-10') != (#:D '2000-03-10')	Returns records where the date is not March 10, 2000.
Return records that have multiple dates in a given field.	And	(#:D '2000-03-10') And (#:D '2003-06-20')	Returns records where the dates are March 10, 2000, and June 20, 2003 for a specified field.
Return records that have one date or another.	Or	(#:D '2000-03-10') Or (#:D '2000-03-11')	Returns records where the date is March 10, 2000, or March 11, 2000.
Return records that contain any values in a given field.	Is Not Null	Is Not Null	Returns records where the field contains any date.
Return records that do not contain any values in a given field.	Is Null	Is Null	Returns records where the field is empty.

Purpose	Operator	Example	Outcome of Example
Return records that have a date that falls within a certain range with respect to the current date.	Now()+ Now()-	Now()-365	Returns records where the date is one year before the current date. The numeric value can be any number of days before or after the current date.*

*If you use an expression to return records where the date is a certain number of days before or after some date, even if you do not specify a time along with the date, the Meridium APM system will assume that you want to return records where the time is exactly 12:00:00 A.M. on that date. For example, if you type the expression (# :d '1997-01-05') +1, the query will return records where the date is January 06, 1997, and the time is exactly 12:00:00 A.M.

Operators to Use with Logical Fields

The following table provides examples of operators that you can use to manipulate values returned in logical fields.

Note that when you type anything in the **Criteria** cell or the **Or** cell, the *is equal to* (=) operator is assumed *at the beginning* of the text unless a different operator is explicitly specified. For example, typing simply `'Y'` in the **Criteria** cell of a logical field causes the Meridium APM system to assume that you want to return records where the value in that field *is equal to true*.

You can type the **=** operator at the beginning of the text, but it is not necessary. If you do not type it, the Meridium APM system will insert it automatically, but it will not be displayed. In addition, the Meridium APM system inserts but does not display the text WHERE or HAVING before the beginning operator (either an implied = operator or an explicit alternative operator). This means that the SQL code will contain either a WHERE clause or a HAVING clause as a result of the text that you type in the **Criteria** cell or the **Or** cell.

Purpose	Operator	Example	Outcome of Example
Return records that have an exact value in a given field.	=	='Y'	Returns records where the field value is **True**.
Return records that do not contain any values in a given field.	Is Null	Is Null	Returns records where the field is empty.
Return records that do not have a specified value in a given field.	Not <> !=	Not 'Y' <> 'Y' != 'Y'	Returns records where the field value is not **True** (i.e., False or Null).

Operators to Use with Numeric Fields

The following table provides examples of operators that you can use to manipulate values returned in numeric fields.

Note that when you type anything in the **Criteria** cell or the **Or** cell, the *is equal to* (=) operator is assumed *at the beginning* of the text unless a different operator is explicitly specified. For example, typing simply **5** in the **Criteria** cell of a numeric field causes the Meridium APM system to assume that you want to return records where the value in that field *is equal to five.*

You can type the **=** operator at the beginning of the text, but it is not necessary. If you do not type it, the Meridium APM system will insert it automatically, but it will not be displayed. In addition, the Meridium APM system inserts but does not display the text WHERE or HAVING before the beginning operator (either an implied = operator or an explicit alternative operator). This means that the SQL code will contain either a WHERE clause or a HAVING clause as a result of the text that you type in the **Criteria** cell or the **Or** cell.

Purpose	Operator	Example	Outcome of Example
Return records that have an exact value in a given field.	=	=10	Returns records where the specified field value is 10.
Return records where the value in a given field is the sum of the specified values.	+	=8+2	Returns records where the specified field value is the sum of 8 and 2, or 10.
Return records where the value in a given field is the difference between the specified values.	-	=11-1	Returns records where the specified field value is the difference between 11 and 1, or 10.
Return records where the value in a given field is the outcome of the specified values after they are divided.	/	=20/2	Returns records where the specified field value is the outcome of 20 divided by 2, or 10.
Return records where the value in a given field is the outcome of the specified values after they are multiplied.	*	=5*2	Returns records where the specified field value is the outcome of 5 multiplied by 2, or 10.

Purpose	Operator	Example	Outcome of Example
Return records that have a value greater than the specified value in a given field.	>	>9	Returns records where the specified field value is greater than 9.
Return records that have a value less than the specified value in a given field.	<	<11	Returns records where the specified value is less than 11.
Return records that have a value greater than or equal to the specified value in a given field.	>=	>=10	Returns records where the specified field value is greater than or equal to 10.
Return records that have a value less than or equal to the specified value in a given field.	<=	<=10	Returns records where the specified field value is less than or equal to 10.
Return records that do not have the specified value in a given field.	Not

!=

<> | Not 10

!= 10

<> 10 | Returns records where the specified field value is not 10. |
Return records that have multiple values in a given field.	And	<5 And >1	Returns records where the specified field values is less than 5 and greater than 1.
Return records that have one value or another in a given field.	Or	5 Or 10	Returns records where the specified field value is 5 or 10.
Return records that have a value similar to the specified value.	Like	Like '10%'	Returns records where the specified field value starts with the digits 10, followed by any number of other digits, such as 234 (10234).

Purpose	Operator	Example	Outcome of Example
Return records that have a value similar to the specified value.	Like	Like '10_'	Returns records where the specified field value starts with the digits 10, followed by any one digit, such as 0 (100).
Return records that do not have a value similar to the specified value.	Not Like	Not Like '10%'	Returns records where the specified field value does not start with the digits 10 followed by any number of additional digits. For example, this would eliminate values 100 and above.
Return records that do not have a value similar to the specified value.	Not Like	Not Like '10_'	Returns records where the specified field value does not start with the digits 10 followed by any one digit. For example, this would eliminate values 100 through 109.
Return records where a given field contains a value that results from a specified grouping.	()	(1000+1)-1	Returns records where the specified field value is equal to 1000 plus 1, or 1001, minus 1, which equals 1000.
Return records that contain any value in a given field.	Is Not Null	Is Not Null	Returns records where the field is not empty.
Return records that do not contain any values in a given field.	Is Null	Is Null	Returns records where the field is empty.
Return records that contain the specified values in a given field.	In	In (5,10,20)	Returns records where the specified field value is 5, 10, or 20.
Return records that do not contain the specified values in a given field.	Not In	Not In (5,10,20)	Returns records where the specified field value is not 5, 10, or 20.

Changing the Case Sensitivity of Expression Values

When you create a query using an Oracle schema, you have the option of making the query criteria case sensitive or case *in*sensitive. By default, new queries created against an Oracle schema are *case sensitive*, which may offer better performance for some queries. If desired, you can modify the default setting to make Oracle queries case insensitive.

> **Note:** Queries constructed against a SQL Server databases are *always* case *in*sensitive. When you are connected to a SQL Server database, the Match Case option described in the following instructions is not available.

The case-sensitivity option that you select for a given query will be applied to any expression values defined for that query. For example, if you define an expression to return only those records that contain the value *Active* in a given field, keeping the default case-sensitive selection will cause the results to include records that contain the value *Active* using the same case that you typed in the **Criteria** or **or** cell. Making the query case *in*sensitive, however, will cause the results to include records with the value *Active* using any capitalization.

To change the case sensitivity of a query:

1. In the Meridium APM Framework, for the query whose case sensitivity you want to change, access the query design.

2. On the **Query Tasks** menu, click the **Options** link.

3. On the submenu, click the **Match Case** option.

   - When the option has a check mark next to it, the expression values will be case sensitive.

   - When this option does not have a check mark next to it, the expression values will not be case sensitive.

Modifying the Query Type

The following instructions provide details on selecting the type of query that you want to create. Additionally, you can use these instructions to modify the query type. For example, if you want to create a Crosstab query from an existing Select query, you can change the query type setting on the Select query from Select to Crosstab.

To modify the query type:

1. In the Meridium APM Framework, for the query whose query type you want to modify, access the query design.

2. On the **Query Tasks** menu, click the **Options** link.

3. On the submenu that appears, select one of the following query types:

    - Select Query
    - Crosstab Query
    - Update Query
    - Append Query
    - Delete Query

> **Note:** You should not create a Delete query from a Select query that contains relationship family sources. If you create a Delete query from a Select query, after you select the relationship family as the target source on the **Target Query Source** dialog box and click the **Select** button, an error message will appear. Additionally, the values in the design canvas and design grid will be removed.

Including and Excluding Fields From the Select Statement

When you create a query, all the fields that you added are selected by default to be included in the SQL select statement. In some cases, you might want to add a field to the design grid so that you can define criteria for that field but not include it in the select statement due to the performance benefits of excluding a field from the select statement.

For example, you might want to configure a query that returns all Pumps installed on a certain date. You could add the Pump ID field and the Asset Installation Date to the query, define criteria to limit the Asset Installation Date to a certain date, and then exclude the Asset Installation Date from the select statement. This would increase performance and still return the records you are interested in viewing.

Via the design grid, you can specify that you want to exclude a field from the SQL select statement.

To exclude a field from the SQL select statement:

1. In the Meridium APM Framework, for the query containing a field that you want to exclude from the SQL select statement, access the query design.

2. In the design grid, in the **Include** cell for the field that you do not want to include in the code, clear the check box.

   > Note: When you clear the **Include** check box for a field, the **Display** check box is automatically cleared for that field. Fields can be displayed in the results only if they are also included in the select statement

   When you run the query, the column for which you cleared the **Include** check box will not appear in the results.

   > Note: If you omit a field from the select statement, you must define criteria, specify sorting order, or add a Total option to that column for that field to be saved with the query code. In other words, if the field is not included in the select statement, it must be included in the SQL code in some other way to be saved with the query. Otherwise, when you open or run the saved query, the excluded field will not be displayed in the query design, the results, or the SQL code.

Determining How Results Will Be Sorted

For any query, you can define criteria to determine how the results will be sorted by default. Sort criteria can be applied via the design grid. When defining sort criteria for a query, note that:

- The sort criteria determines the *default* sort order for query results. After the results are displayed, users can modify the sort order on-the-fly in the results grid.
- If NO sort criteria has been defined, the query results will not be sorted in any particular order by default. Users can still modify the sort order on-the-fly in the results grid.
- Any sort criteria that is defined for numeric columns in a query will be applied to *stored* values (vs. displayed values) regardless of whether the query is running in formatted or unformatted mode.
- Unlike the sorting that you can apply on-the-fly to the results, sorting preferences that you define in the query design are stored with the query itself and are applied each time you run the query.

For example, according to the following image, the Asset Installation Date will be sorted in Ascending order.

Field	[Asset ID]	[Asset Description]	[Asset Installation Date]
Alias	Asset ID	Asset Description	Asset Installation Date
Table	[Pump]	[Pump]	[Pump]
Sort			Ascending
Sort Index			1

These sort options will affect the query results as shown in the following image.

Asset ID	Asset Description	Asset Installation Date
PMP-0380	SPARE AT TK2227	1/1/1940
PMP-1383	D-5 REFLUX	1/1/1945
PMP-1384	D-5 REFLUX	1/1/1945
PMP-1390	G-1A SPARE	1/1/1945
PMP-2323	SUMP OIL	1/1/1945
PMP-1386	WAX TO STORAGE	1/1/1945
PMP-2345	G-2 SOLVENT CHARGE	1/1/1946
PMP-2612	#2 FURFRUAL	1/1/1948
PMP-2613	#2 FURFURAL	1/1/1948

Note: The sort order that you specify within the query definition will be saved as the default layout for the results. The default layout will be overridden if a different layout is defined via the results grid and saved with the query.

To determine how query results will be sorted:

1. In the Meridium APM Framework, for the query whose results you want to sort, access the query design.

2. In the design grid, in the **Sort** cell for the field that you want to use as the primary sort, select **Ascending** or **Descending**. Note that the **Sort Index** cell displays a 1, indicating that this field will be used as the primary sort value. In the following example, the Asset Installation Date will be sorted in Ascending order.

3. If you want to set a secondary sort, in the **Sort** cell of the appropriate field, select **Ascending** or **Descending**. Note that the Sort Index cell displays a 2, indicating that this field will be used as the primary sort value.

4. Continue in this way until you have defined all the desired sort values.

   After you run the query, the columns for which you defined a sorting preference will be sorted in the specified order.

Changing the Page Size of the Results

To change the page size of the results:

1. In the Meridium APM Framework, for the query whose page size you want to change, access the query design.

2. On the **Query Tasks** menu, click the **Options** link, and then click **Change Page Size**.

   The **Change Page Size** dialog box appears.
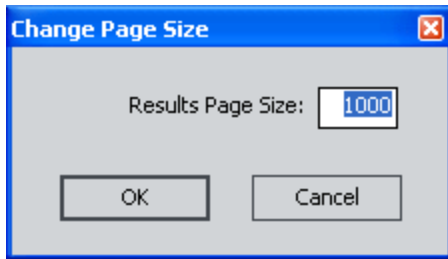
3. In the **Results Page Size** text box, type the number of rows you want the results to contain per page. The default is **1000**.

4. Click **OK**.

   > **Note:** If you type a value greater than or equal to 10,000, a message appears, asking if you are sure you want to set the selected page size although it may consume a considerable amount of time and system resources. If you want to set the selected page size, click **Yes**.

5. On the **Query Tasks** menu, click the **Run Query** link.

   The query results are displayed according to your selection in the **Change Page Size** dialog box. For example, if you typed **50** in the **Change Page Size** dialog box, and the query returns 100 records, the results will contain two pages, and each page will display 50 records.

About Displaying Custom Text Instead of Field Values

By default, the values displayed in each column in the results are the values in the fields of records included in the results. You can modify the default behavior, however, if you want all cells in a column to display the same text.

For example, if your query returns the Asset Installation Date for all Pump records, the values in the Asset Installation Date column will be the installation dates that are stored in the Pump records in your database. You could, instead, make the column display custom text instead of the actual date. For example, the installation dates might fall within a given time period, such as January 2005 and December 2005, so you might want the text to read **Installed in 2005** to indicate the time frame instead of the actual date.

This means that instead of the results looking like these:

Asset ID	Asset Description	Asset Installation Date
PMP-8468	KNOCK ENG COOL H20	1995-01-01 00:00:00
PMP-8434	CUTTER STOCK	1995-01-01 00:00:00
PMP-8432	CUTTER STOCK	1995-01-01 00:00:00
PMP-8410	VAC TAR RECYCLE/TRAY WASH	1995-01-01 00:00:00
PMP-8279	CAUSTIC RECIRCULATION PUMP	1995-01-01 00:00:00

...they would look like these:

Asset ID	Asset Description	Installed in 2005'
PMP-8279	CAUSTIC RECIRCULATION PUMP	Installed in 2005
PMP-8410	VAC TAR RECYCLE/TRAY WASH	Installed in 2005
PMP-8432	CUTTER STOCK	Installed in 2005
PMP-8434	CUTTER STOCK	Installed in 2005
PMP-8468	KNOCK ENG COOL H20	Installed in 2005

In most cases, you will probably want to use this feature if your query results contain a hyperlink and you want to customize the text of the hyperlink. For example, if your query returns all Pump records, you might include the Asset ID, Asset Description, and Asset Installation Date in the results. By default, the columns will display the value stored in the Asset ID, Asset Description, and Asset Installation field of each record.

Then, you might decide to add a hyperlink to the Asset ID field that will open in the Record Manager each record from the results. Now, suppose you want the hyperlinks to display some text *other than* the Asset ID. For example, you might want the hyperlink to read **Open in Record Manager**, as shown in the following image.

Asset ID	Asset Description	Asset Installation Date
Open in Record Manager	G-2 SOLVENT CHARGE	1/1/1946 12:00:00 AM
Open in Record Manager	SPARE COOLING H20 PUMP	1/1/1954 12:00:00 AM
Open in Record Manager	RESID TRANSFER	1/1/1992 12:00:00 AM
Open in Record Manager	DEBUTANIZER OVHD PRODUCT	1/1/1955 12:00:00 AM
Open in Record Manager	TK720 HDC-GAS OIL FEED	1/1/1992 12:00:00 AM
Open in Record Manager	TRANSFER-(ISO- BUTANE) TK 2159	1/1/1956 12:00:00 AM
Open in Record Manager	ABSORBER TOPINTERCOOLER	1/1/1956 12:00:00 AM

To modify the values displayed in a column, you will need to modify the **Field** cell for the desired column in the design grid.

> ŒHint: If you modify the Field cell for a column that contains Group By in the Total cell, an error message will appear when you run the query. If you receive this error, try changing Group By to Min, Max, or Expression.

Modifying the Field Cell

To modify the value in the Field cell:

1. In the Meridium APM Framework, for the query for which you want modify the value in the **Field** cell, access the query design.

2. In the design grid, locate the column that represents the field whose value you want to define.

3. In the **Field** cell for that column, delete the existing value.

4. In the empty **Field** cell, type the text that you want to appear in the results, surrounding the text with single quotation marks. Note that this text will appear in *each* cell in the column. For example, if you want the text to read **Open in Record Manager**, type `'Open in Record Manager'`

Field	'Open in Record Manager'	[Asset Description]
Alias	Asset ID	Asset Description
Table		[Pump]

When you run the query, the text you typed will appear in each cell in the column representing the selected field. In this example, the **Asset ID** column contains the text **Open in Record Manager**.

Asset ID	Asset Description
Open in Record Manager	SPARE AT TK2227
Open in Record Manager	G-1A SPARE
Open in Record Manager	D-5 REFLUX
Open in Record Manager	WAX TO STORAGE
Open in Record Manager	SUMP OIL

Note that the text appears as a hyperlink if a hyperlink has been defined for the field.

About Limiting the Number of Results That are Returned

For certain queries, you might want to limit the number of results that are returned such that only the first *n* number of records are returned based upon a specific set of criteria.

> Note: This feature can be used only with Select queries.

For example, suppose that you want to view the 20 most expensive equipment items according to total cost, where the value in the Breakdown Indicator field is set to *True*.

In this case, you would create the query by including the families, fields, and criteria that are shown in the following image.

Notice that the Total Cost field is included, and the query is configured to show the sum of all total cost values, grouped by Equipment ID. For example, suppose that the piece of equipment with the Equipment ID 000000000001060839 is linked to two Work History records, where the value in the Total Cost field in each record is *720,000* and *301*, respectively. In this case, rather than showing two *separate* rows in the results for each Work History record that is linked to this Equipment record, the results will contain *one* row for this Equipment record, and the **Total Cost** column will display the *sum* of the individual values in the Total Cost field in each Work History record. In other words, in the row containing this Equipment record, the **Total Cost** column will display the value *720,301*.

In addition, the query is configured to show the results in descending order according to total cost.

Based on this query configuration, if you were to run the query at this point, you would see ALL Equipment records that meet the query criteria. In the following image, you can see that 5,693 records are returned when the query is run.

Equipment ID	Taxonomy Category Descripti	Location ID	Asset Tech ID	Equipment Short Description	Total Cost
00000000001060839	Fixed Asset	MRD-ROA-REFN-RF104-B0001-104	B0001-104	BOILER - STEAM - NORTH BOILER GAS FIRED	4072201
00000000001060840	Fixed Asset	MRD-ROA-REFN-RF104-B0002-104	B0002-104	BOILER - STEAM - RCCS SOUTH BOILER GAS FIRED	2605956
00000000001067911	Fixed Asset	MRD-ROA-REFN-RF114-B0004-114	B0004-114	BOILER - STEAM - NO. 4 GAS FIRED BOILER	2558822
00000000001068100	Fixed Asset	MRD-ROA-REFN-RF114-F0108-114	F0108-114	PRESSURE VESSELS W/ INTERNALS	1618185
00000000001105535	Fixed Asset	MRD-ROA-REFN-RF116-PIPE-116	PIPE-116	A P I SEPARATOR - PIPING UPGRADES	1603982
00000000001105624	Fixed Asset	MRD-ROA-REFN-RF117-PIPE-P-117	PIPE-P-117	PROCESS PIPING REPAIRS	1519249
00000000001067915	Fixed Asset	MRD-ROA-REFN-RF114-B0010-114	B0010-114	BOILER - STEAM - NO. 10 BOILER GAS FIRED	1328407
00000000001016654	Fixed Asset	MRD-ROA-REFN-RF031-D0001-031	D0001-031	VESSEL - TOWER	1276601
00000000001050134	Rotating Asset	MRD-ROA-REFN-RF091-GC0033-091	GC0033-091	COMPRESSER POSITIVE DISPLACEMENT	1183415
00000000001050133	Rotating Asset	MRD-ROA-REFN-RF091-GC0032-091	GC0032-091	COMPRESSOR - POSITIVE DISPLACEMENT AND	1182611
00000000001005862	Fixed Asset	MRD-ROA-REFN-RF009-B0007-009	B0007-009	BOILER - STEAM - PACKAGE BOILERS GAS FIRED	1046595
00000000001052628	Fixed Asset	MRD-ROA-REFN-RF093-E0009B-093	E0009B-093	AIRCOOLED FIN FAN	1037599
00000000001035576	Fixed Asset	MRD-ROA-REFN-RF067-E0026A-067	E0026A-067	EXCHANGERS, U-BEND 2026E0026A	903794
00000000001035504	Fixed Asset	MRD-ROA-REFN-RF067-B0002-067	B0002-067	HEATER - FIRED	897354
00000000001067916	Fixed Asset	MRD-ROA-REFN-RF114-B0012-114	B0012-114	STEAM BOILERS - FIRED	866680
00000000001103594	Fixed Asset	MRD-ROA-REFN-RF073-PIPE-073	PIPE-073	C5/C6 ISOMERIZATION - PIPING UPGRADES	863108
00000000001013915	Rotating Asset	MRD-ROA-REFN-RF026-GC0001-026	GC0001-026	COMPRESSOR - POSITIVE DISPLACEMENT AND	720301
00000000001056817	Fixed Asset	MRD-ROA-REFN-RF097-E0011J-097	E0011J-097	EXCHANGER - FIN FAN CONDENSER	618560
00000000001105527	Fixed Asset	MRD-ROA-REFN-RF114-PIPE-W-114	PIPE-W-114	STEAM SYSTEMS WATER PIPING REPAIRS	609038
00000000001100332	Fixed Asset	MRD-ROA-REFN-RF009-PIPE-S-009	PIPE-S-009	STEAM PIPING REPAIRS	603179
00000000001032834	Fixed Asset	MRD-ROA-REFN-RF064-D0004-064	D0004-064	VESSEL - TOWER	597919

Page: 1 of 6   5,693 records found   «« First   « Previous   Next »   Last »»   Page Size: 1000

Based on this scenario, however, you want to view only the top *20* most expensive pieces of equipment. So, you still need to refine the query to meet this need. To do so, in the query design, you would specify *20* as the desired limit. After doing so, when you ran the query again, you would see only the first 20 Equipment records returned by the query, as shown in the following image.

Equipment ID	Taxonomy Category Descripti	Location ID	Asset Tech ID	Equipment Short Description	Total Cost
00000000001060839	Fixed Asset	MRD-ROA-REFN-RF104-B0001-104	B0001-104	BOILER - STEAM - NORTH BOILER GAS FIRED	4072201
00000000001060840	Fixed Asset	MRD-ROA-REFN-RF104-B0002-104	B0002-104	BOILER - STEAM - RCCS SOUTH BOILER GAS FIRED	2605956
00000000001067911	Fixed Asset	MRD-ROA-REFN-RF114-B0004-114	B0004-114	BOILER - STEAM - NO. 4 GAS FIRED BOILER	2558822
00000000001068100	Fixed Asset	MRD-ROA-REFN-RF114-F0108-114	F0108-114	PRESSURE VESSELS W/ INTERNALS	1618185
00000000001105535	Fixed Asset	MRD-ROA-REFN-RF116-PIPE-116	PIPE-116	A P I SEPARATOR - PIPING UPGRADES	1603982
00000000001105624	Fixed Asset	MRD-ROA-REFN-RF117-PIPE-P-117	PIPE-P-117	PROCESS PIPING REPAIRS	1519249
00000000001067915	Fixed Asset	MRD-ROA-REFN-RF114-B0010-114	B0010-114	BOILER - STEAM - NO. 10 BOILER GAS FIRED BOILER	1328407
00000000001016654	Fixed Asset	MRD-ROA-REFN-RF031-D0001-031	D0001-031	VESSEL - TOWER	1276601
00000000001050134	Rotating Asset	MRD-ROA-REFN-RF091-GC0033-091	GC0033-091	COMPRESSER POSITIVE DISPLACEMENT	1183415
00000000001050133	Rotating Asset	MRD-ROA-REFN-RF091-GC0032-091	GC0032-091	COMPRESSOR - POSITIVE DISPLACEMENT AND ROTARY	1182611
00000000001005862	Fixed Asset	MRD-ROA-REFN-RF009-B0007-009	B0007-009	BOILER - STEAM - PACKAGE BOILERS GAS FIRED	1046595
00000000001052628	Fixed Asset	MRD-ROA-REFN-RF093-E0009B-093	E0009B-093	AIRCOOLED FIN FAN	1037599
00000000001035576	Fixed Asset	MRD-ROA-REFN-RF067-E0026A-067	E0026A-067	EXCHANGERS, U-BEND 2026E0026A	903794
00000000001035504	Fixed Asset	MRD-ROA-REFN-RF067-B0002-067	B0002-067	HEATER - FIRED	897354
00000000001067916	Fixed Asset	MRD-ROA-REFN-RF114-B0012-114	B0012-114	STEAM BOILERS - FIRED	866680
00000000001103594	Fixed Asset	MRD-ROA-REFN-RF073-PIPE-073	PIPE-073	C5/C6 ISOMERIZATION - PIPING UPGRADES	863108
00000000001013915	Rotating Asset	MRD-ROA-REFN-RF026-GC0001-026	GC0001-026	COMPRESSOR - POSITIVE DISPLACEMENT AND ROTARY	720301
00000000001056817	Fixed Asset	MRD-ROA-REFN-RF097-E0011J-097	E0011J-097	EXCHANGER - FIN FAN CONDENSER	618560
00000000001105527	Fixed Asset	MRD-ROA-REFN-RF114-PIPE-W-114	PIPE-W-114	STEAM SYSTEMS WATER PIPING REPAIRS	609038
00000000001100332	Fixed Asset	MRD-ROA-REFN-RF009-PIPE-S-009	PIPE-S-009	STEAM PIPING REPAIRS	603179

Page: 1 of 1 20 records found «« First « Previous Next » Last »» Page Size: 1000

If you are using a SQL Server database, when you sort the query by a field and then configure the query to return a limited number of records, if more than one record contains the same value in the field by which you sorted the query, those records will be displayed in a random order, relative to one another. For instance, if two Equipment records were linked to Work History records with the *same* total cost, each time you ran the query, those Equipment records would be displayed in a random order, relative to one another. In other words, the first time you ran the query, the Equipment record with the ID 00000000001060839 might appear *above* the Equipment record with the ID 00000000001060840. The second time you ran the query, however, those Equipment records might appear in the reverse order. This limitation applies to SQL Server databases only. For customers using a SQL Server database and this feature, you will want to construct your query in a way that prevents this limitation from causing inaccurate results to be returned.

Defining the Number of Results That Should be Returned

To define the number of results that should be returned:

1. In the Meridium APM Framework, for the query for which you want to define the number of results that are returned, access the query design.

2. On the **Query Tasks** menu, click the **Options** link, and then click **Limit Results Set**.

   The **Limit Results Set** dialog box appears.
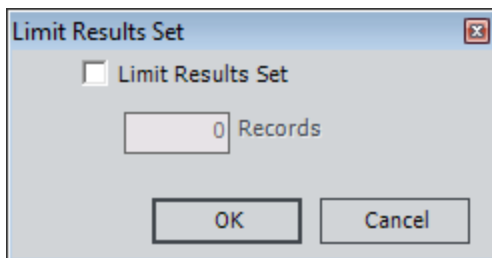
3. Select the **Limit Results Set** check box.

4. In the **Records** text box, type the number of records that you want the query to return. The default number is *0*.

5. Click **OK**.

   > **Note:** If you type a value greater than or equal to 10,000, a message appears, asking if you are sure you want to specify this number although it may consume a considerable amount of time and system resources. If you want to specify this number, click **Yes**.

6. On the **Query Tasks** menu, click the **Run Query** link.

   The query results are displayed according to your selection in the **Limit Results Set** dialog box. For example, if you typed *20* in the **Limit Results Set** dialog box, the query will return only the first 20 records that meet the other query criteria.

Removing the Limit on the Number of Results That Should be Returned

To remove the limit on the number of results that should be returned:

1. In the Meridium APM Framework, for the query from which you want to remove the limit on the number of results that are returned, access the query design.

2. On the **Query Tasks** menu, click the **Options** link, and then click **Limit Results Set**.

   The **Limit Results Set** dialog box appears.
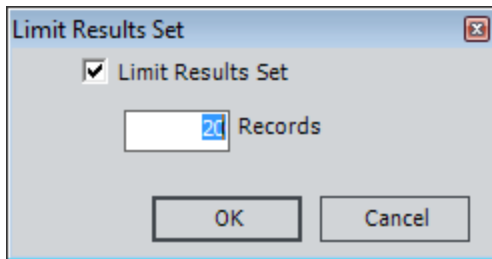
3. Clear the **Limit Results Set** check box.

   The **Records** text box is disabled, and the number that appeared in it is removed.

4. Click **OK**.

5. On the **Query Tasks** menu, click the **Run Query** link.

   The query results are displayed, and the number of results that are returned is no longer limited.

Displaying Unique Records Only

You can specify that results of a query return only unique records. For example, if you design a query to return all types of Air Cooled Heat Exchangers, you might get 10 results, but there might be only two different types of Air Cooled Heat Exchangers within those results, such as Air Cooled Heat Exchanger and Cooling Tower. If you specify that the query return only records where the asset type is unique, you will see only two results: one Air Cooled Heat Exchanger result and one Cooling Tower result.

Note that uniqueness is defined at the query level, not the field level. This means that if the same query that returns two results with different asset types is also designed to return the asset status, and you specify that the query return only unique records, you will see records where the asset type and status are unique. So, you could see the results shown in the following image, where there are active Air Cooled Heat Exchangers, inactive Air Cooled Heat Exchangers, and active Cooling Towers in the database.

Asset Type	△	Asset Status
Air Cooled Heat Exchanger		Active
Air Cooled Heat Exchanger		Inactive
Cooling Tower		Active

To specify that the query return unique records only:

1. In the Meridium APM Framework, for the query that should display unique records only, access the query design.

2. On the **Query Tasks** menu, click the **Options** link.

3. On the submenu, select the **Unique Records Only** option.

   When you run the query, your results will reflect only unique records.

About Formatted and Unformatted Mode

You can run Select queries in two different modes:

- **Formatted Mode:** Causes the results to display *formatted* values rather than stored values. In other words, the values displayed in the results will be formatted based upon any format rules or criteria defined for the fields included in the query.

- **Unformatted Mode:** Causes the results to display *stored* values rather than formatted values. In order words, any format rules that have been defined will not be applied, and the results will display values exactly as they exist in the database.

Consider an example where the System Code Table *Priority* contains the System Codes listed in the following table and is configured to display descriptions only.

ID	Description
1	Very High
2	High
3	Medium
4	Low
5	Very Low

Now, assume that your database contains the Task family, which contains a Priority field, and also assume that a Valid Values rule has been applied to this field so that it displays a list of values from the *Priority* System Code Table. Because this System Code Table is set up to display descriptions only, the available values for this field will be Very High, High, Meridium, Low, and Very Low. When a user selects one of these values, the corresponding numeric *ID* will be stored in the field.

When you create a Select query on the Task family, you can choose to run it in formatted or unformatted mode. If you create a query that includes the Task ID, Task Type, and Priority fields and run it in *unformatted* mode, the results might look something like those shown in the following image.

Task ID	Task Type	Priority
9245873	INTERNAL_VISUAL	2
9856781	INTERNAL_VISUAL	3
9876512	EXTERNAL_VISUAL	5
9874570	INTERNAL_VISUAL	1
9987458	EXTERNAL_VISUAL	2
9856478	INTERNAL_VISUAL	1
9784586	EXTERNAL_VISUAL	5
9457813	EXTERNAL_VISUAL	5
9132456	EXTERNAL_VISUAL	5
9846781	EXTERNAL_VISUAL	4
9213458	INTERNAL_VISUAL	4
9458712	EXTERNAL_VISUAL	3
9632587	INTERNAL_VISUAL	2
9658742	EXTERNAL_VISUAL	4

Notice in this image that *numeric* values are displayed in the **Priority** column. These are the System Code *ID* values that are stored in the Priority field of these records. If you run the same query in *formatted* mode, the results would look like those shown in the following image.

Task ID	Task Type	Priority
9245873	INTERNAL_VISUAL	High
9856781	INTERNAL_VISUAL	Medium
9876512	EXTERNAL_VISUAL	Very Low
9874570	INTERNAL_VISUAL	Very High
9987458	EXTERNAL_VISUAL	High
9856478	INTERNAL_VISUAL	Very High
9784586	EXTERNAL_VISUAL	Very Low
9457813	EXTERNAL_VISUAL	Very Low
9132456	EXTERNAL_VISUAL	Very Low
9846781	EXTERNAL_VISUAL	Low
9213458	INTERNAL_VISUAL	Low
9458712	EXTERNAL_VISUAL	Medium
9632587	INTERNAL_VISUAL	High
9658742	EXTERNAL_VISUAL	Low

Notice that the **Priority** column now displays the System Code *descriptions* that correspond to the stored IDs. In other words, the results now show the *formatted* values rather than the stored values.

By default, newly created Select queries will run in *unformatted* mode. You can change the mode and save it with the query so that the next time you run the query, it will use the mode that you last saved. Note that the output mode option is available only for Select queries.

> **Note:** If you select the **Show Totals** option, the query behaves as if it were running in unformatted mode even if you select to run it in formatted mode.

Modifying the Output Mode of Select Queries

To modify the output mode of a Select query:

1. In the Meridium APM Framework, for the Select query whose output mode you want to change, [access the query design](#).

2. On the **Query Tasks** menu, click **Options**, point to **Mode**, and then click the desired mode: **Formatted** or **Unformatted**.

3. On the **Query Tasks** menu, click the **Run Query** link.

   The query is run in the selected mode, and the results are displayed.

4. On the **Common Tasks**menu, click the **Save Query** link to save the query. The selected mode is saved with the query. The next time you run the query, it will run using the mode that was selected when it was last saved.

About Sort Options for Formatted Queries

When you run a Select query in formatted mode, the query results will display formatted values rather than stored values. For example, a field that stores a currency value might store the value *100*, but if a format rule has been applied to that field, the value might be displayed as *$100.00* in a query configured to run in formatted mode.

When you execute a query in formatted mode where the displayed value differs from the stored value, sorting on the displayed value versus the stored value may produce different results. Therefore, when you run a Select query in formatted mode, you have the option of sorting the results on the formatted (i.e., displayed) value or the stored values.

Consider an example of a query on the Task family. If we choose to sort our formatted query on the *displayed* values, when we sort the **Priority** in ascending order, the results would look like those shown in the following image.

Task ID	Task Type	Priority	Δ
9245873	INTERNAL_VISUAL	High	
9987458	EXTERNAL_VISUAL	High	
9632587	INTERNAL_VISUAL	High	
9846781	EXTERNAL_VISUAL	Low	
9213458	INTERNAL_VISUAL	Low	
9658742	EXTERNAL_VISUAL	Low	
9856781	INTERNAL_VISUAL	Medium	
9458712	EXTERNAL_VISUAL	Medium	
9874570	INTERNAL_VISUAL	Very High	
9856478	INTERNAL_VISUAL	Very High	
9876512	EXTERNAL_VISUAL	Very Low	
9784586	EXTERNAL_VISUAL	Very Low	
9457813	EXTERNAL_VISUAL	Very Low	
9132456	EXTERNAL_VISUAL	Very Low	

Notice that the values in the **Priority** column are sorted in ascending alphabetical order according to the displayed value. This sort order does not correspond to the actual priority order of the tasks, where tasks with a Very High priority should appear at the top of the list, and tasks with a Very Low priority should appear at the bottom of the list. Alternatively, if you were to sort the results based upon the *stored* value, the results would look like those shown in the following image.

Task ID	Task Type	Priority	Δ
9874570	INTERNAL_VISUAL	Very High	
9856478	INTERNAL_VISUAL	Very High	
9245873	INTERNAL_VISUAL	High	
9987458	EXTERNAL_VISUAL	High	
9632587	INTERNAL_VISUAL	High	
9856781	INTERNAL_VISUAL	Medium	
9458712	EXTERNAL_VISUAL	Medium	
9846781	EXTERNAL_VISUAL	Low	
9213458	INTERNAL_VISUAL	Low	
9658742	EXTERNAL_VISUAL	Low	
9876512	EXTERNAL_VISUAL	Very Low	
9784586	EXTERNAL_VISUAL	Very Low	
9457813	EXTERNAL_VISUAL	Very Low	
9132456	EXTERNAL_VISUAL	Very Low	

Notice that the Priority column is now sorted in ascending order by the *stored* numeric value. Using this sort option, the sort order corresponds to the actual task priority, where the tasks with a Very High priority appear at the top of the list.

Changing the Sort Option

> **Note:** The sort option described in this topic applies only to *numeric* columns in Select queries that are configured to run in *formatted* mode. Unformatted queries will always be sorted on *stored* values, which in unformatted mode match the displayed values. Additionally, this option affects *only* the [on-the-fly sorting that can be applied in the results](#). Any sort criteria defined within the query itself will be applied to *stored* values.

To set the sort option for a Select query:

1. In the Meridium APM Framework, open a Select query in the query design.

2. On the **Query Tasks** menu, click **Options**, point to **Mode**, and confirm that the **Formatted** option is selected. When this option is selected, the **Sort Results Grid by Actual Value** option will be enabled.

3. On the **Options** menu, click the **Sort Results Grid by Actual Value** option to toggle it to the desired setting.

   - When a check mark appears to the left of this option, it is turned on, meaning that your results will be sorted by *stored* values.
   - When a check mark does not appear to the left of this option, it is turned off, meaning that your results will be sorted by *displayed* values.

4. On the **Query Tasks** menu, click the **Run Query** link.

   The query is run, and the results are displayed. In the results grid, sort the results to ensure that the sort option had the desired effect.

5. On the **Common Tasks** menu, click the **Save Query** link.

   The query is saved, along with the sort setting that you defined.

Accessing the Report Viewer Page

To access the Report Viewer page:

- In the Meridium APM Framework, <u>open an existing report</u>.

  -or-

  In the Meridium APM Framework, <u>create a new report</u>.

  The **Report Viewer** page appears. The Catalog item caption is displayed in the title bar to identify the report.

Overview of the Report Viewer Page

The **Report Viewer** page contains the following items:

- **Report workspace:** The report workspace contains a [toolbar](#) and the actual report.
- **Task menus:** The **Report Viewer** page contains the [Report Tasks menu](#) and the **Common Tasks** menu.

Using the **Report Viewer** page, you can perform the following tasks:

- [View and modify report properties](#).
- [Export reports](#).
- [Modify reports](#).
- [View and edit the query or dataset design](#).

About the Report Workspace

The report workspace contains [a toolbar](#) and the report itself. In addition, a right-click menu is available.

In the following image, the report workspace is outlined in red.
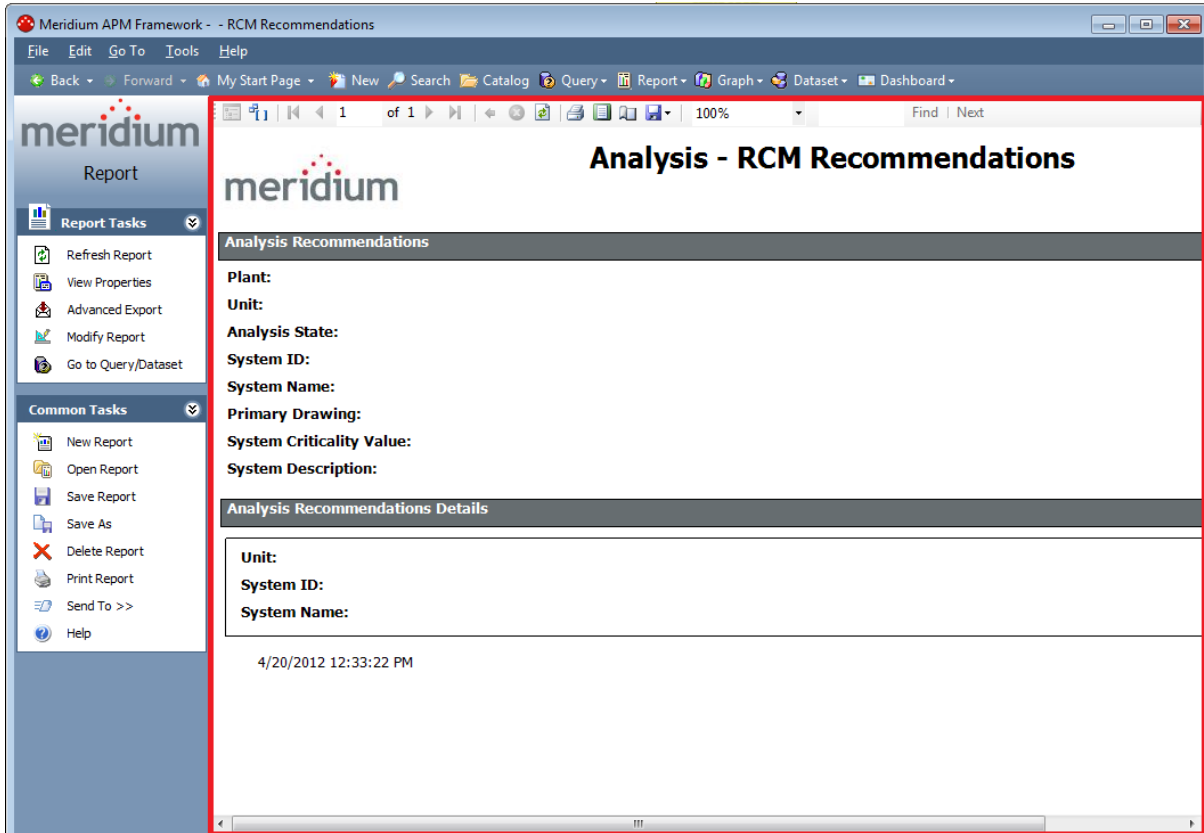
About the Toolbar

The toolbar that you see when you are viewing a report on the **Report Viewer** page offers various options that you can use to navigate, print, or export a report. Note that you cannot modify a report on the **Report Viewer** page. If you want to modify a report, you can click the **Modify Report** link on the **Report Tasks** menu, which will open the report designer.

Report Tasks Menu

The **Report Tasks** menu on the **Report Viewer** page contains the following links:

- **Refresh Report:** Reloads the report to ensure that it is displaying the most current information, including any changes you may have made. Note that if the underlying query contains a prompt, the **Enter Query Parameters** dialog box appears, where you will need to specify the desired parameters before reloading the report.

- **View Properties:** Displays the **Report Properties dialog box**, which displays information about the report, including the name, description, location, and date last updated. This link is enabled only after you save the report.

- **Advanced Export:** Displays the **Save As** dialog box, which lets you save the report to an external location in a variety of formats. Note that this link is disabled if the Business Intelligence Development Studio is not installed on your Meridium APM client machine.

- **Modify Report:** Launches the report designer, where you can modify the currently displayed report.

  > **Note:** When you click this link, a message is displayed asking if you want to include a subreport. If you click the **Yes** button, you can browse to the desired subreport and then open the main report and its subreports in the report designer. If you click the **No** button, you will be directed automatically to the report designer, where you can modify the currently displayed report.

  This link is enabled only if the Business Intelligence Development Studio is installed on the Meridium APM client machine.

- **Go to Query/Dataset:** Opens the query or dataset from which the report was generated.

Common Tasks Menu

The **Common Tasks** menu on the **Report Viewer** page contains the following links:

- **New Report:** Displays the **Create New Report** dialog box, where you can choose to base the new report on a query, dataset, or external report.

  Although the link is enabled for all users, you will not be able to proceed with creating a report unless the Business Intelligence Development Studio is installed.

- **Open Report:** Displays the **Open an existing Report** dialog box, where you can open an existing report.

- **Save Report:** Saves the current report.

- **Save As:** Displays the **Save Report As** dialog box, where you can save the current report with a different name or to a different location.

- **Delete Report:** After asking for confirmation, deletes the current report and any associated subreports.

- **Print Report:** Displays the **Print** dialog box, where you can print the current report.

  > **Note:** When you print a report, the time that appears on the printed report is the time at which the report was printed. However, when you view a report in a print preview layout, the time that appears on the report is the time at which the report was last run.

- **Send To:** Displays a submenu with options that let you .

- **Help:** Displays the context-sensitive Help topic for the current page.

About Creating a New Report

You can create a new report based upon an existing query or dataset or based upon an external SQL Server Reporting Services report. To create reports, Microsoft SQL Server Business Intelligence Development Studio must be installed on the Meridium APM Framework machine. This tool provides a user interface that you can use to develop and modify reports. Throughout this documentation, we refer to this tool as the *report designer*.

After you create a report, you can save it to the Catalog in any folder to which you have been granted the necessary permissions.

To create a report, you can:

- Use the **Build New** option on the **Reports** menu (accessible via the Meridium APM Framework toolbar).

- Use the **Create a Report** link on the **Other Tasks** menu (accessible via the query results).

- Use the **Create a new report** link on the **Other Tasks** menu (accessible via the **Dataset Editor** page).

> **Note:** The options listed above are enabled only if the Business Intelligence Development Studio is installed on the Meridium APM Framework machine. In addition, the Meridium APM Adapter for SSRS is also needed on Meridium APM Framework workstations for building reports that are based upon a Meridium APM data source. If the Meridium APM Adapter for SSRS is not installed on the Meridium APM Framework machine, when you build a report, you will not be able to preview the report output. For this reason, we recommend that both the Business Intelligence Development Studio and the Meridium APM Adapter for SSRS be installed on the Meridium APM Framework.

Note that Meridium APM supports the use of subreports, which are individual reports that provide data that is then included in a larger main report. The main report displays the combination of all data included in its subreports.

Creating a Report Based on a Query or Dataset

To create a new report based on a query or dataset:

1. In the Meridium APM Framework, on the Meridium APM Framework toolbar, click **Reports**, and then click **Build New**.

   The **Create New Report** dialog box appears.

2. On the **Create New Report** dialog box, select either the **Meridium APM Query** option or the **Meridium APM Dataset** option, and click **OK**.

   The **Select a Query or Dataset for Report** dialog box appears.

3. Navigate to the query or dataset that you want to use to create the report, select it, and click the **Open** button.

   A message appears, asking if you want to include a subreport.

4. If you do not want to include a subreport, click the **No** button.

   -or-

   If you want to include a subreport, click the **Yes** button. The following instructions assume that you want to include a subreport.

   The **Choose a Subreport** dialog box appears.

5. Navigate to the report that you want to use as the subreport, select it, and click the **Open** button.

   The **Selected Subreports** dialog box appears, displaying in the list the subreport that you selected on the **Choose a Subreport** dialog box and asking if you want to include another subreport.

6. If you want to add another subreport, click the **Yes** button, and select the desired subreport from the **Choose a Subreport** dialog box.

   -or-

   If you do not want to add another subreport, click the **No** button.

   > **Note:** If you selected a query that contains a prompt, the **Enter Query Parameters** dialog box will appear, where you must specify the query parameters before the report will be created.

   After you have added the desired subreport(s), the report designer appears.

   > **Note:** When the report designer is displayed, the following folder is created automatically: **<boot drive>:\Documents and Settings\<User>\MeridiumTempReport\tmpreports**, where **<User>** is a folder representing the current Windows user. Report files are created and stored in this folder until you a) exit out of the report designer without saving the report or b)

save the report, exit out of the report designer, and save the report to the Catalog.

7. Proceed through the report designer to create the report as desired.

8. Save the report using the **Save** option. Do not use the **Save As** option.

> ŒHint: Using the **Save As** option results in the creation of a copy of the current file. If you choose to save the current report with a new name, the results of the file with the current name will continue to be displayed. For this reason, we recommend that you save any changes using the **Save** command.

9. Exit out of the report designer.

   The report is passed into the Meridium APM Framework application and appears on the **Report Viewer** page.

10. Save the report to the Meridium APM Catalog. When you do so, the report will be published to the SQL Server Report Server. You can view the report on the Report Server using the Report Manager.

About Using an External Template to Create a Report

Meridium APM offers an option that allows you to create a report that is:

- Based on a report that you created inside the report designer and then saved outside of the Meridium APM Framework application.
- Based on a report that you created from a Meridium APM query or dataset and have saved only outside of Meridium APM, not to the Catalog. For example, suppose you have begun the process of building a report based on a query but you need several days to complete the report. You can save the report to a drive and then access it from within the Meridium APM Framework application later. You can decide to save it to the Catalog whenever it is complete.

The basic workflow to create a report from an external template is outlined in the following list:

1. Create the report, either against a Meridium APM query or dataset or natively inside of the report designer.

2. Save the report file to a location that is accessible from the Meridium APM Framework machine.

3. In the Meridium APM Framework application, choose to create a report from an external report using the **Create New Report** dialog box.

4. Save the report to the Catalog. The report is then accessible through the Meridium APM Framework application and can be displayed in the Report Viewer, where all standard report functions are available.

In general, we recommend that you build reports based on Meridium APM queries or datasets instead of against the underlying database because internal data tables are subject to change from one Meridium APM version to the next. Meridium APM queries and datasets, on the other hand, are much more stable and are upgraded as necessary. In some cases, however, a report should be built against the underlying database directly instead of on top of a query or dataset. For example, reports that use highly complex database queries should be built against the database.

Reports based on external reports are transportable across databases as long as the tables (families) used in the report exist in both databases. For example, if a report is written against the Test database, a user connecting to the Meridium APM Production database could use the same report to return different data from the same family.

This documentation explains how to create a report based on an external report after the report has already been created using the report designer. We do not provide instructions for using the report designer.

> **Note:** Reports written directly within the report designer should not use Meridium APM as the data source type. If you want to create a report against a Meridium APM

query or dataset, you should begin the report-creation process in the Meridium APM Framework application. Alternatively, you can query a Meridium APM database directly by using the SQL Server or Oracle data source type.

Creating a Report Based on an External Report

To create a new report based on an external report:

1. After creating the report using the report designer, save the report file to a location that is accessible from the Meridium APM Framework machine.

2. In the Meridium APM Framework, on the Meridium APM Framework toolbar, click **Reports**, and then click **Build New**.

   The **Create New Report** dialog box appears.

3. Select the **External Template** option, and click **OK**.

   The **Open** dialog box appears.

4. Navigate to the report you want to open, and click the **Open** button.

   > **Note:** The report must be a .RDL file.

   The external report appears on the **Report Viewer page**.

5. On the **Report Tasks** menu, click the **Modify Report** link.

   The report designer is launched, and you can use the external report as a template to create a new report.

   > **Note:** When the report designer is displayed, the following folder is created automatically: **<boot drive>:\Documents and Settings\<User>\MeridiumTempReport\tmpreports**, where **<User>** is a folder representing the current Windows user. Report files are created and stored in this folder until you a) exit out of the report designer without saving the report or b) save the report, exit out of the report designer, and save the report to the Catalog.

6. After you have modified the report as desired, save the report using the **Save** command. Do not use the **Save As** command.

   > Œ**Hint:** Using the **Save As** command results in the creation of a copy of the current file. If you choose to save the current report with a new name, the results of the file with the current name will continue to be displayed. For this reason, we recommend that you save any changes using the **Save** command.

   The new report is passed into the Meridium APM Framework application and appears on the **Report Viewer page**.

7. To save the report to the Catalog, on the **Common Tasks** menu, click the **Save Report** link. In addition, the report is published to the Report Server. You can view the report on the Report Server using the Report Manager.

Opening Reports in the Meridium APM Framework

The following instructions explain how to open an existing report from the Meridium APM Framework toolbar. You can also open an existing report by opening it directly from the Catalog.

> **Note:** If you are viewing a report based on a query, the report may or may not display the most current information in the database. If additional fields have been *added* to the query and the report was not [modified](#) accordingly after the query fields were added, the report will not reflect the additional fields. If, however, fields have been *removed* from the underlying query, the report will not display those fields even if the report was not modified after the query fields were removed.

To open a report:

1. In the Meridium APM Framework, on the Meridium APM Framework toolbar, click **Reports**, and then click **Open Existing**.

    The **Open an existing Report** dialog box appears.
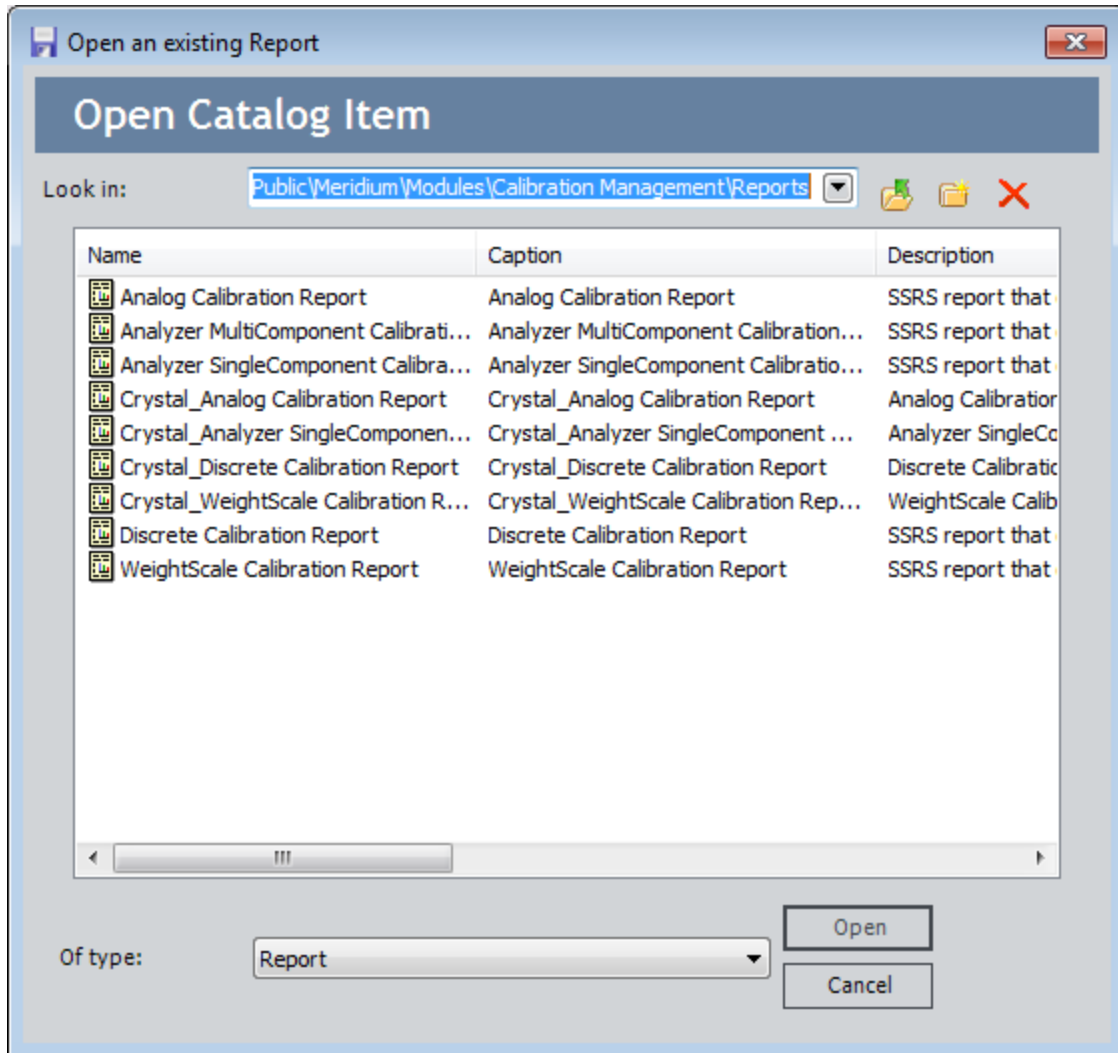
2. Navigate to the report that you want to open, select the report, and click the **Open** button.

The report appears on the [Report Viewer page](). Note that if you opened a report that has not yet been published to the SQL Server Report Server, the report is published to the server when you open it. In addition, if the report contains subreports that are stored in the same Catalog folder as the main report, the subreports will also be published to the server when you open the main report.

> **Note:** Depending on the number of subreports that are included in a report, it may take some time to appear. The more subreports that are included in the report, the longer it will take for the report to appear.

Viewing SQL Server Reporting Services Reports Outside of Meridium APM

The SQL Server Report Manager is a tool that is provided with SQL Server Reporting Services that lets you manage reports. Because you can use the Report Manager to view Meridium APM reports, we provide some information about this tool in this documentation. Additional details about the Report Manager can be found in the Help system that is accessible from the Report Manager tool.

Reports are first deployed to the SQL Server Report Server when they are saved to the Meridium APM Catalog. This means that after a report exists in the Meridium APM Catalog, it can be viewed in Report Manager.

> Œ Hint: If you rename a report in the Meridium APM Catalog, you will need to open the report before the new name will be reflected in Report Manager.

The Report Manager is accessible from any web browser, either on the Report Server itself or remotely via the following URL:

http:\\\<Report Server Name>\\<Report Manager Virtual Directory>

...where:

- **<Report Server Name>** is the name of the Report Server.
- **<Report Manager Virtual Directory>** is the name of the virtual directory that was created for the Report Manager.

All Meridium APM reports can be accessed through the Report Manager. Reports are organized on the Report Server using a folder structure that corresponds to the Meridium APM Catalog folder structure.

All Meridium APM reports will be stored automatically under the **Home\Meridium** directory on the Report Server.

- The **Home** directory is created automatically when you install SQL Server Reporting Services.
- The **Home\Meridium** directory is created automatically the first time a report is published to the Report Server. Publishing occurs when a user creates a new report or accesses an existing report from the Meridium APM Catalog. If needed, you can create this directory manually for the purpose of granting permissions.

Within the **Home\Meridium** directory, subdirectories are created automatically as needed, as reports are created and viewed within the system. The following format is used for creating directories to store report files:

\\Home\Meridium\<Data Source Name>\<Meridium APM Catalog Path>

...where:

- **<Data Source Name>** is the name of the data source that was used for creating the report.

  > **Note:** Any special characters that are included in the data source name will not be included in the corresponding folder on the Report Server as SQL Server Reporting Services does not allow special characters in folder paths. Special characters will be removed from the data source name automatically when the folder is first created.

- **<Meridium APM Catalog Path>** is the full path to the report in the Meridium APM Catalog.

For example, suppose that in the database DGT_ALL_323, the report **Shell and Tube HE Report** is stored in the Meridium APM Catalog folder \\Public\Meridium. In this case, you would find the report on the Report Server in the directory **\\Home\Meridium\DGT_ALL_323\Public\Meridium**, as shown in the following image.
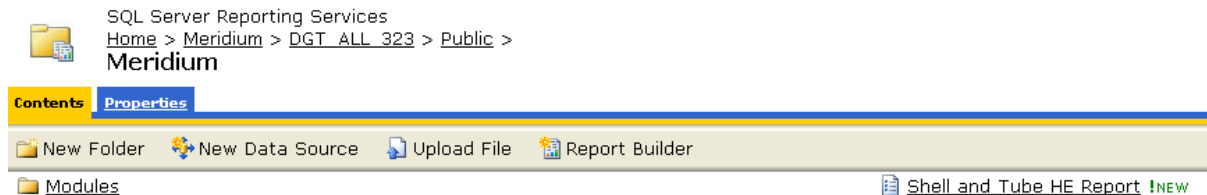
When you attempt to open a Meridium APM report in Report Manager, you will be prompted to provide a user ID and password. You can supply any Meridium APM user ID and password with the Meridium APM security privileges. The results that are displayed in the Report Manager should look the same as the results that are displayed when you open the report in the Meridium APM Framework application.

SQL Server Report Manager should be used only to view Meridium APM reports. You should not create or modify reports directly through Report Manager. When a user modifies an existing report in the Meridium APM Framework application and re-saves it to the Meridium APM Catalog, the report will be redeployed automatically to the Report Server. Any modifications that have been made on the Report Server via Report Manager will be overwritten when the report is redeployed from the Meridium APM Framework Application.

About Modifying Reports

You can think of a report as the combination of the database content included in the report and the way in which the content is laid out, including column headers, footers, separators, and so on. Meridium APM does not provide a default layout for reports. Upon their initial creation, you will see a blank report that you will need to format and populate as desired.

In the Meridium APM Framework application, the **Report Viewer** page provides a means of viewing the report, including the database content it includes and the layout and appearance of that content.

If the Business Intelligence Development Studio is installed on your computer, when you are viewing a report in the Report Viewer, you can access and modify the report.

Note that if you modify a subreport, the next time that you open a main report that includes that subreport, the modifications to the subreport will be reflected in the main report.

If desired, you can modify the underlying query. If you choose to do so, note that:

- If an underlying query is modified by adding new fields, the report based on that query will be automatically updated the next time that it is viewed, provided that the new fields have been added to the report.

- If fields are removed from a query, the report based on that query will be automatically updated, and no manual changes to the report are needed.

- If an alias is modified in an underlying query and the associated report contains field, the values that are stored in that field will not be displayed in the report the next time that it is viewed. You will need to modify the report or build a new report from the query in order to see the values that are stored in that field.

Modifying Reports

To modify a report:

1. In the Meridium APM Framework, on the **Report Viewer** page, on the **Report Tasks** menu, click the **Modify Report** link.

   The report designer is launched.

2. Modify the report as desired.

3. Save your changes using the **Save** command. Do not use the **Save As** command.

   > Œ**Hint:** Using the **Save As** command results in the creation of a copy of the current file. If you choose to save the current report with a new name, the results of the file with the current name will continue to be displayed. For this reason, we recommend that you save any changes using the **Save** command.

4. Exit out of the report designer.

   The modified report appears on the **Report Viewer page**.

Viewing and Modifying the Underlying Query or Dataset

You can view and edit the query or dataset that was used to create a report.

To view or modify the query or dataset for a report:

- In the Meridium APM Framework, on the **Report Viewer** page, on the **Report Tasks** menu, click the **Go To Query/Dataset** link.

  The query or dataset appears. After the query or dataset is displayed, you can modify the query or modify the dataset as desired.

Changing the Underlying Query or Dataset for a Report

When you create a report that is based upon a query or dataset, the Catalog path and item name of that query or dataset are associated with the report. If you move the underlying query or dataset to a different location or rename it, you will need to update the report to reference the new location or item name.

> Note: These instructions assume that the content of the query or dataset are the same and simply the Catalog path or item name has changed.

To change the underlying query or dataset for a report:

1. In the Meridium APM Framework, on the **Report Viewer** page, on the **Report Tasks** menu, click the **Modify Report** link.

   > Note: When you click this link, a message is displayed asking if you want to include a subreport. If you click the**Yes**button, you can browse to the desired subreport and then open the main report and its subreports in the report designer. If you click the**No**button, you will be directed automatically to the report designer.

   The report designer is launched.

2. Change the data source for the report to reference the new query or dataset. For details, see the Business Intelligence Development Studio Help.

3. Save the report in the report designer, and exit the application.

   The report workspace refreshes.

4. On the **Common Tasks** menu, click the **Save Report** link.

   The report properties are updated with the new path and item name.

Adding a Subreport to an Existing Report

To add a subreport to an existing report:

1. In the Meridium APM Framework, open the report to which you want to add a sub-report.

2. On the **Report Viewer** page, on the **Report Tasks** menu, click the **Modify Report** link.

   A message appears, asking if you want to include a subreport.

3. Click the **Yes** button.

   The **Choose a Subreport** dialog box appears.

4. Navigate to the report that you want to use as the subreport, select it, and click the **Open** button.

   The **Selected Subreports** dialog box appears, displaying the subreport that you selected in step 4 appears in the list.

5. If you want to add another subreport, click the **Yes** button, and then repeat step 4.

   -or-

   If you do not want to add another subreport, click the **No** button.

   The report designer appears.

6. Proceed through the report designer to create the report as desired.

7. Save the report using the **Save** command. Do not use the **Save As** command.

   > ŒHint: Using the **Save As** command results in the creation of a copy of the current file. If you choose to save the current report with a new name, the results of the file with the current name will continue to be displayed. For this reason, we recommend that you save any changes using the **Save** command.

   The report is passed into the Meridium APM Framework application and appears on the Report Viewer page.

Viewing and Modifying Report Properties

Reports are stored as items in the Meridium APM Catalog. As with all Catalog items, you can modify the graph properties via the Catalog itself. Additionally, via the **Report Viewer** page, you can view the Catalog item properties and modify the caption and description.

To view the properties associated with the current report:

1. In the Meridium APM Framework, on the **Report Viewer** page, on the **Report Tasks** menu, click the **View Properties** link.

   The **Report Properties** dialog box appears.

2. If desired, in the **Caption** text box, modify the caption of the report.

3. If desired, in the **Description** text box, modify the description of the report. You can click the [globe icon] button to manage translations for that string.

   The remaining properties are view-only.

4. Click **OK**.

   Your changes are saved.

About Exporting Reports

You can export reports using the following features:

- The **Report Tasks** menu.
- The right-click menu that is available when you are viewing a report.
- The toolbar on the **Report Viewer** page.

Using any of these features, you can export reports in the following formats:

- XML file
- CSV (comma delimited)
- .TIF file
- Adobe Acrobat
- Web archive
- Microsoft Excel

Exporting a Report Using the Report Tasks Menu

Using the **Report Tasks** menu, you can export a report in the following formats:

- XML file
- CSV (comma delimited)
- .TIF file
- Adobe Acrobat
- Web archive
- Microsoft Excel

To export a report using the Report Tasks menu:

1. In the Meridium APM Framework, on the **Report Viewer** page, on the **Report Tasks** menu, click the **Advanced Export** link.

   The **Save As** dialog box appears.

2. In the file structure, navigate to the location in which you want to save the report.

3. In the **File name** field, type a name for the report.

4. In the **Save as type** field, select the format in which you want to save the report.

5. Click the **Save** button.

   The report is saved to the selected location as the selected file type.

Exporting a Report Using the Right-click Menu

Using the right-click menu while viewing a report, you can export in the following formats:

- XML file
- CSV (comma delimited)
- .TIF file
- Adobe Acrobat
- Web archive
- Microsoft Excel

To export a report using the right-click menu:

1. In the Meridium APM Framework, on the **Report Viewer** page, right-click anywhere in the report workspace.

   The right-click menu appears.

2. On the shortcut menu, point to **Export**, and then select the format in which you want to export the report.

   The **Exporting** dialog box appears, followed by the **Save As** dialog box.

3. In the file structure, navigate to the location in which you want to save the report.

4. In the **File name** list, type a name for the report.

5. Click the **Save** button.

   The report is saved to the selected location as the selected file type.

Exporting a Report Using the Report Viewer Toolbar

Using the toolbar on the **Report Viewer** page while viewing a report, you can export a report in the following formats:

- XML file
- CSV (comma delimited)
- .TIF file
- Adobe Acrobat
- Web archive
- Microsoft Excel

To export a report using the toolbar on the Report Viewer page:

1. On the toolbar, click the [icon] button.

   A drop-down list box appears.

2. Select the format in which you want to export the report.

   The **Exporting** dialog box appears, followed by the **Save As** dialog box.

3. In the file structure, navigate to the location in which you want to save the report.

4. In the **File name** list, type a name for the report.

5. Click the **Save** button.

   The report is saved to the selected location as the selected file type.

About Associating Reports with Families

Each entity family can be associated with one or more reports, which will be available for selection when a user chooses to print a record in that family in the Record Manager. The main benefit of using family reports is the ability to display the record in a customized layout that has been designed using a report.

> **Note:** Before you can associate reports with families in your database, you must first create the reports themselves and any necessary queries and save them in the Catalog.

While any report can be associated with a family, the intent is for administrative users to build reports based upon queries that contain an entity key prompt. When a record is opened in the Record Manager, the entity key from the current record will be passed in to the query prompt, thereby limiting the report content to that of the current record.

When a user chooses to print a record in the Record Manager, a check will be performed to determine whether a report is linked to the current family. If none are available, the standard datasheet printing will be performed. If one or more reports have been associated with the family, the **Select Report to Print** dialog box will appear.

From here, the user will be able to:

- Accept the default report selection.
- Choose a different report from the list of reports that have been associated with the family.
- Choose to print the standard datasheet layout.

Accessing the List of Family Reports

To view the reports associated with a family:

1. In the Configuration Manager, in the **Entity Family/Relationship Family** pane, click the **Entity Family** tab.

2. Select the family whose reports you want to view.

3. In the **Tasks** section, click the **Manage Family Reports** link.

   The **Family Reports** window appears.

From the **Family Reports** window, you can associate a family with a report and delete family reports.

Associating a Report with a Family

By associating a family with a report, that report will be available for selection when a user chooses to print a record in that family.

To associated a family with a report:

1. In the Configuration Manager, on the **Family Reports** window, click the **Add** button.

   The **Select Report to Link** dialog box appears.

2. Navigate to the desired report, select it, and click the **Open** button.

   > **Note:** If you select *All* in the **Of type** list, you will be able to browse to and select non-report Catalog items. Note, however, that only reports can be associated with families. If you associate a non-report item with a family, when you save your selections, an error message will be displayed, and you selections will not be saved.

   The report is added to the **Family Reports** list.

3. If desired, add additional reports by repeating the previous steps.

4. When you are finished adding reports, click **OK** to close the **Family Reports** window.

5. On the **Meridium APM Configuration Manager** main window, at the bottom of the **Physical Storage** section, click the **Save** link.

   A message appears, indicating that the family was updated successfully.

Removing Reports from a Family

These instructions provide details on removing a report from a family. Removing a report removes the association between the report and the family. It does not delete the report from the Catalog.

> **Note:** When a report is associated with a family, it cannot be deleted from the Catalog in the Meridium APM Framework application. If you need to delete from the Catalog a report that is associated with a family, you must first remove the report from the family.

To remove a report from a family:

1. In the Configuration Manager, on the **Family Reports window**, select the report that you want to remove.

2. Click the **Remove** button.

   A confirmation message appears, asking if you really want to remove the report.

3. Click the **Yes** button.

   The report is removed from the list of family reports.

Designating a Report as the Default Report

When a report is as the default report for a family, that report will be selected automatically when a user chooses to print a record in the Record Manager. The user will be able to change the selection, if desired. But designating a report as the default report can be useful if you are associating multiple reports with a family, one of which will be used most often. If you associate more than one report with a family, we recommend that you designate one of those reports as the default report.

To designate a report as the default report:

1. In the Configuration Manager, on the **Family Reports window**, in the row that contains the report you want to make the default, select the **Default** check box.

2. Click **OK**.

    The selected report becomes the default report for the print job.

Saving Reports

After you [create a report](#), you can save it so that you or other users can access it again later. Reports are saved as Catalog items to the Meridium APM Catalog. After a report has been saved to the Meridium APM Catalog, whenever you make changes to it, you will save to re-save the report to retain your changes.

To save a report to the Meridium APM Catalog:

1. While viewing the report on the **Report Viewer** page, on the **Common Tasks** menu, click the **Save Report** link.

   If the report has already been saved, any changes that you have made will be saved to the current Catalog folder and with the existing Catalog item properties. If the report has not yet been saved, the **Save Report** dialog box will appear, as shown in the following image. The following instructions assume that you are saving the report for the first time.
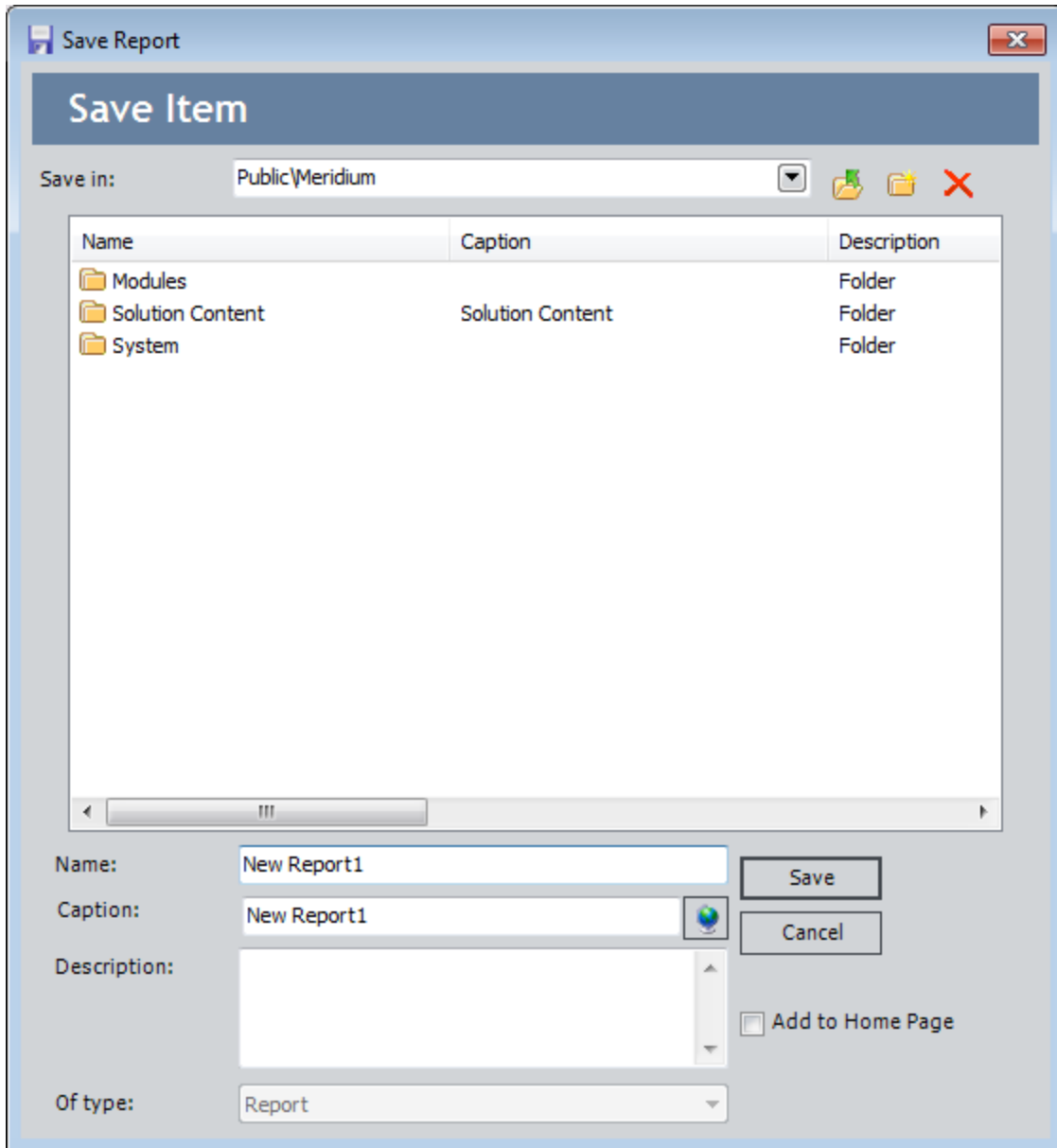
Note: If you are viewing a saved report on the **Report Viewer** page and you click the **Save As** link on the **Common Tasks** menu, the **Save Report** dialog box will appear so that you can save a *copy* of the current report with a different name or to a different location. You can use the following instructions to complete that task, as the process is the same as saving a report for the first time.

2. Navigate to the Catalog folder in which you want to save the report.

3. In the **Name** text box, type a name for the report. The name is required and must be unique to the Catalog folder in which you are saving the report.

4. If desired, modify the Value in the **Caption** text box. Additionally, you can click the ![globe] button to manage translations for that string.

5. If desired, type a description of the dashboard in the **Description** text box. This is not required to save the report.

6. If desired, select the **Add to Home Page** check box if you want to add a link to the report to a Home Page. This is not required to save the report.

7. Click the **Save** button.

    If you:

    - Selected the **Add to Home Page** check box, the report is saved to the Catalog, and the Send to Home Page - Select Home Page dialog box is displayed.

    Depending upon your membership in the MI Power User Role Security Group and whether or not you are a Super User, you can use the **Send to Home Page - Select Home Page** dialog box provide a link to the dashboard to your Personal Home Page, a Group Home Page, or the Personal Home Page for a particular Security User.

    - Did not select the **Add to Home Page** check box, or you have completed the process of adding the dashboard to a Home Page, the **Save Report** dialog box closes, and the report is saved to the Catalog.

Deleting Reports

To delete a report from the database:

1. In the Meridium APM Framework, [open the report](#) that you want to delete.

2. On the **Report Viewer** page, on the **Common Tasks** menu, click the **Delete Report** link.

   A confirmation message appears, asking if you really want to delete the report.

3. Click the **Yes** button.

   The report is deleted from the database. Note the following about deleting reports:

- If you delete a report from the Meridium APM database, the report is also deleted from the SQL Server Report Server.

- If you delete a subreport, the next time that you open a main report that includes that subreport, a message will appear, indicating that the subreport cannot be displayed.

Aspects of the R Script Editor

The **R Script Editor** page appears when you open an existing R Script or create a new R Script. Throughout this documentation, we refer to this page simply as the **R Script Editor**.

To function properly, the **R Script Editor** requires Revolution Analytics to be installed on the Meridium Application Server.

The **R Script Editor** is divided into the *editor* and *mapping* sections. If you are a viewing a new R Script in the **R Script Editor** that you have not yet saved, the heading above the editor section will be *<New R Script>* until you save the R Script. After you save the R Script, *<New R Script>* will be replaced with the caption of the R Script as it is stored in the Catalog item.

For example, in the following image, the *Base Thinning Damage Factor* R Script is displayed in the **R Script Editor**. The editor and mapping sections are outlined in blue and green, respectively.
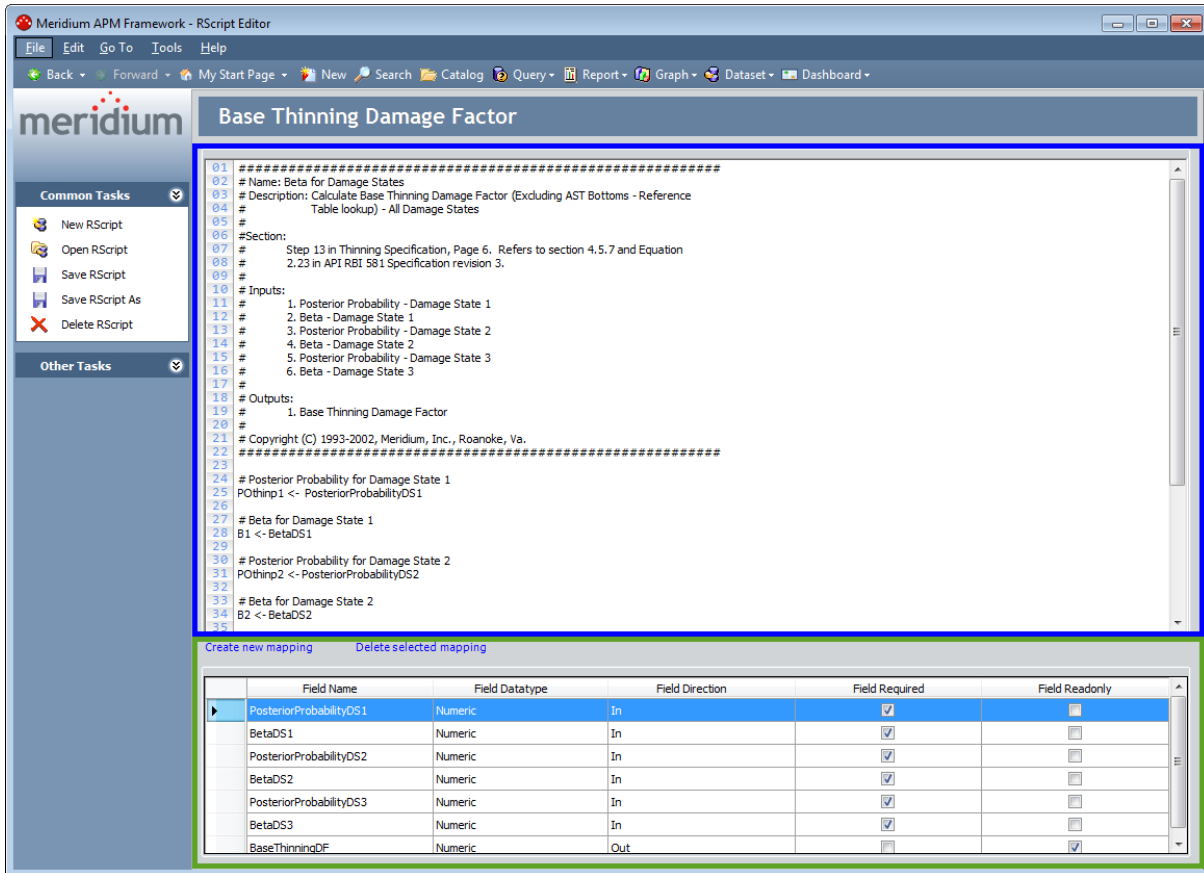
Common Tasks Menu
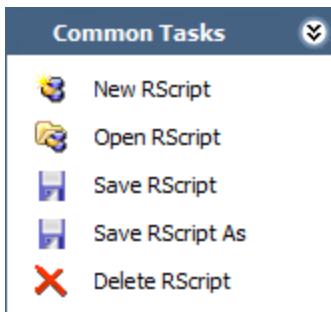
The **Common Tasks** menu on the **R Script Editor** contains the following links:

- **New R Script:** Closes the current R Script, and displays empty editor and mapping sections. The header will be set to *<New R Script>*.

  > **IMPORTANT:** If you have R code in the editor section, or if you have made changes to an existing R Script, when you click the **New R Script** link, you will *not* be prompted to save your changes. Any unsaved work will be lost.

- **Open R Script:** Displays the Open an existing R Script dialog box, where you can open an existing R Script.

- **Save R Script:** Saves the current R Script.

- **Save R Script As:** Displays the **Save R Script As** dialog box, where you can save the current report with a different name or to a different location.

- **Delete R Script:** After asking for confirmation, deletes the current R Script.

  > **Note:** After deleting an R Script, the **Save R Script**, **Save R Script As**, and **Delete R Script** links are disabled. To re-enable the links, you must create or open an R Script.

About Creating R Scripts

You can create R Scripts that are as simple or as complex as you need them to be, depending on the data that you want the script to calculate. For instance, a simple R Script may calculate and return a value based only on two inputs. A more complex R Script might utilize ten inputs, and output ten different values. R Scripts are intended to expand the function of Policies. This documentation does not cover the process of developing R code.

The process of creating an R Script includes the following steps:

1. Create a new R Script.

2. Enter R code in the **R Script Editor**. Code can be typed or pasted into the editor section.

3. Create mappings for the inputs.

4. Save the R Script.

Creating R Scripts

To Create an R Script:

1. In the Meridium APM Framework, in the **Tools** Menu, click **R Scripts**, and then click **Add New**.
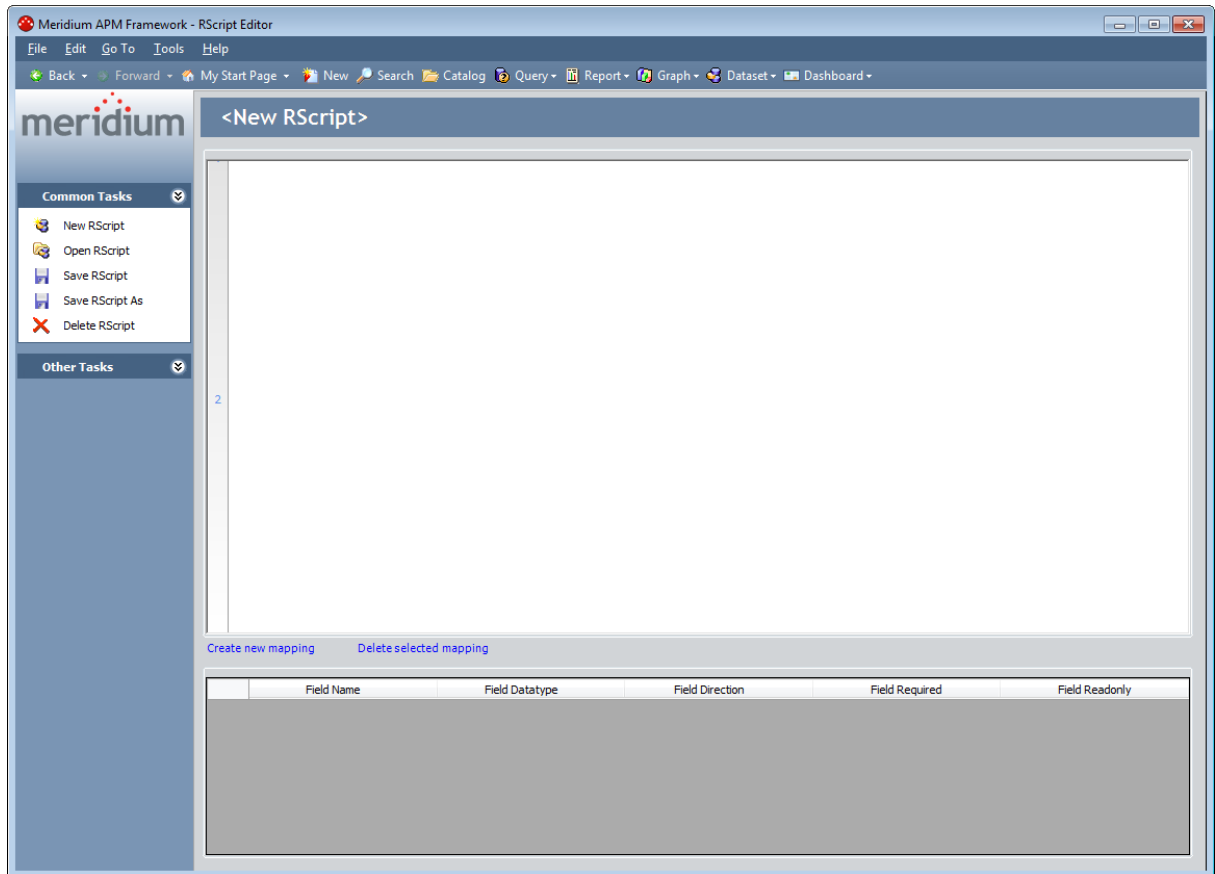
2. In the editor section, type or paste your R code.

3. For each variable that will require a value, create an *In* mapping.

4. For each variable that you want to return a value out of the R Script, create an *Out* mapping.

5. Save the R Script to the Meridium APM Catalog.

   The R Script is saved to the Catalog. It can now be used to create [R-Script nodes] in Policies.

Creating Mappings in R Scripts
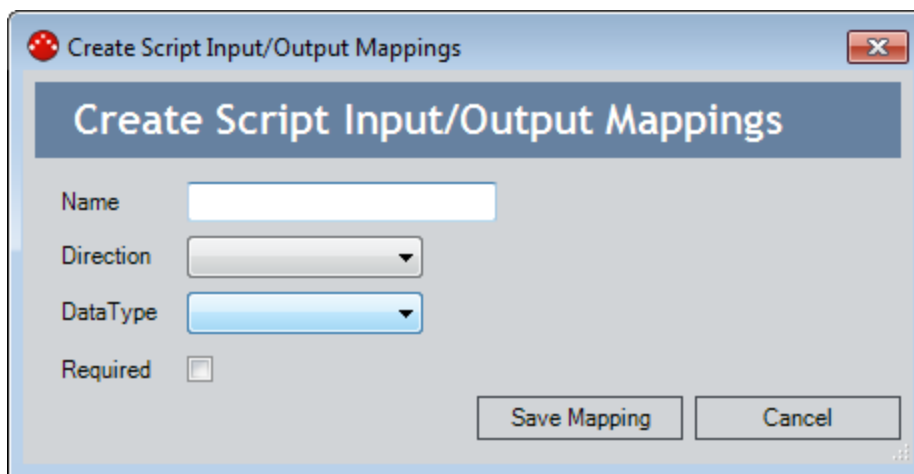
When you define a mapping in an R Script, you are creating an input or output that corresponds to a variable in the R Script. In addition, these mappings show up as fields in the **Properties** window when using an R-Script node in a policy diagram. For example, in the case of mapping inputs, this allows you to pass values from other linked nodes in a policy diagram into the variables in your R Script by specifying what node will correspond to what variable. Conversely, in the case of mapping an output, you define what variable in the R Script will pass a value out of the R-Script node and to another node in a policy diagram.

While you can create a mapping for a variable that is not in the R Script, for mappings to function correctly the variable you specify in the mapping *must* be defined in the code of the R Script. This procedure assumes you are creating a mapping for a defined variable.

To Create a Mapping in an R Script:

1. While viewing the R Script in the R Script Editor, in the mapping section, click the **Create new mapping** link.

   The **Create Script Input/Output Mappings** dialog box appears.

2. In the **Name** text box, type one of the variables that exists in the R Script.

3. In the **Direction** list box, select *In* or *Out*.

   > **Note:** This documentation refers to mappings as *In* mappings or *Out* mappings, based on the value in the **Direction** list box.

4. In the **DataType** list box, select *Numeric* or *Character*.

5. If desired, click the **Required** check box.
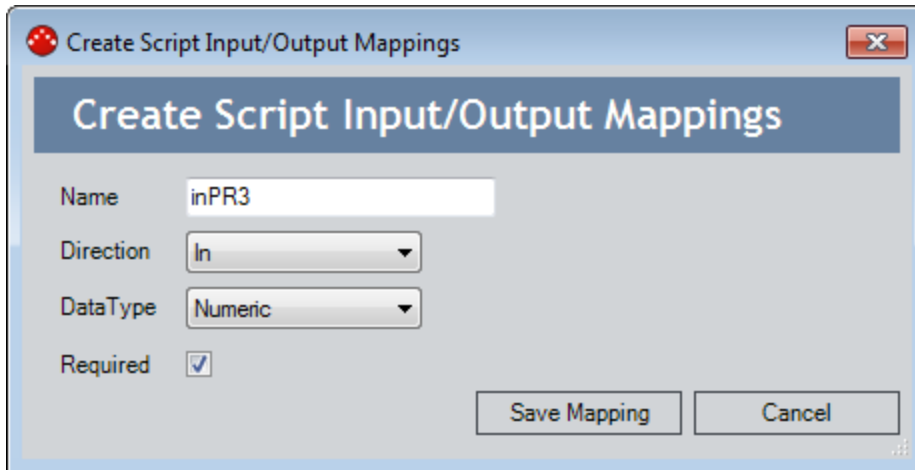
6. Click the **Save Mapping** button.

The mapping is saved. In the mapping section, a new row appears.

If you created an *Out* mapping, in the new row, the **Field Readonly** check box is selected automatically.

Modifying Mappings in R Scripts

To Modify a Mapping in an R Script:

1. While viewing the R Script in the R Script Editor, in the [mapping section](#), double-click on the row corresponding to the mapping you want to modify.

   The **Create Script Input/Output Mappings** dialog box appears.
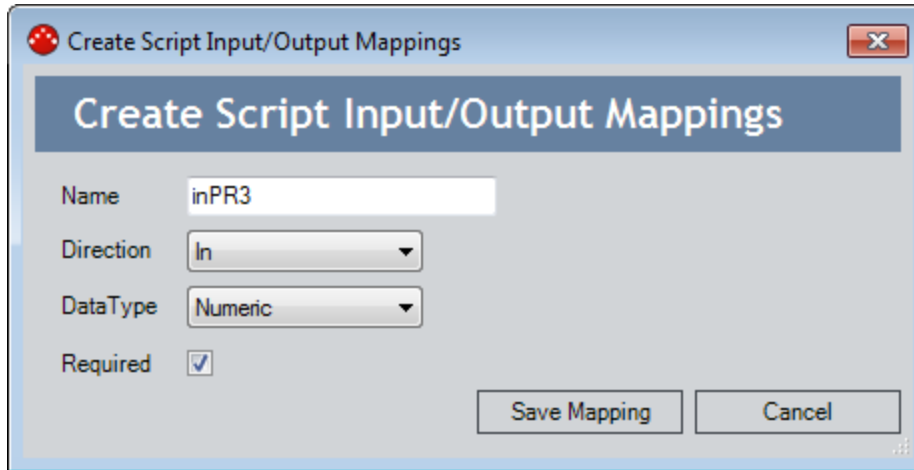
2. If desired, make changes to the mapping.

3. Click the **Save Mapping** button.

   The mapping is saved. In the mapping section, the row corresponding to the mapping is updated to reflect the changes.

Deleting Mappings in R Scripts

To Delete a Mapping in an R Script:

1. In the mapping section, click the row corresponding to the mapping you want to delete.

   The row is highlighted.

	Field Name	Field Datatype	Field Direction	Field Required	Field Readonly
	inPR3	Numeric	In	☑	☐
▶	outPRDval	Numeric	Out	☐	☑

2. In the mapping section, click the **Delete selected mapping** link.

   The mapping is deleted. In the mapping section, the row corresponding to the mapping is removed.

	Field Name	Field Datatype	Field Direction	Field Required	Field Readonly
▶	inPR3	Numeric	In	☑	☐

Modifying R Scripts

To Modify an R Script:

1. In the Meridium APM Framework, in the **Tools** Menu, click **R Scripts**, and then click **Open Existing**.
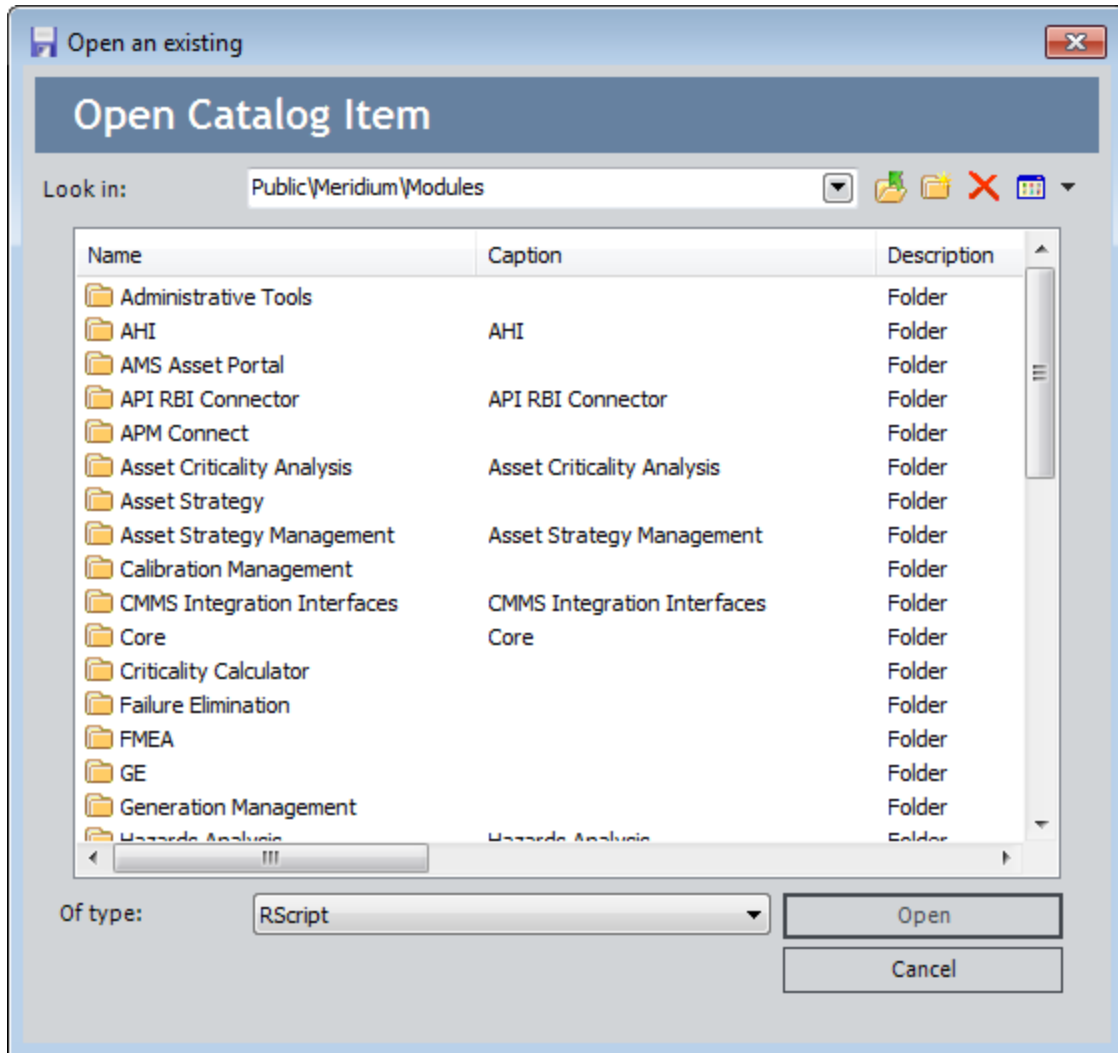
2. Navigate to the R Script that you want to open, select the R Script, and click the **Open** button.

   The R Script appears in the **R Script Editor**.

3. When you are finished making changes, on the **Common Tasks** menu, click **Save R Script**.

   The R Script is saved. If you have added new mappings to the R Script, when you next open a policy diagram containing an R-Script node that uses this R Script, new fields will be added to the **Properties** window. Conversely, if you removed mappings, the corresponding fields will also be removed from the **Properties** window.

Saving R Scripts

After you [create an R Script](#), you can save it so that you or other users can access it again later, and use it in policies. R Scripts are saved as Catalog items to the Meridium APM Catalog. After an R Script has been saved to the Meridium APM Catalog, whenever you make changes to it, you must save the R Script again to retain your changes.

To save an R Script to the Meridium APM Catalog:

1. While viewing the R Script in the **R Script Editor**, on the **Common Tasks** menu, click the **Save R Script** link.

   If the R Script has already been saved, any changes that you have made will be saved to the current Catalog folder and with the existing Catalog item properties. If the R Script has not yet been saved, the **Save R Script As** dialog box will appear, as shown in the following image. The following instructions assume that you are saving the R Script for the first time.
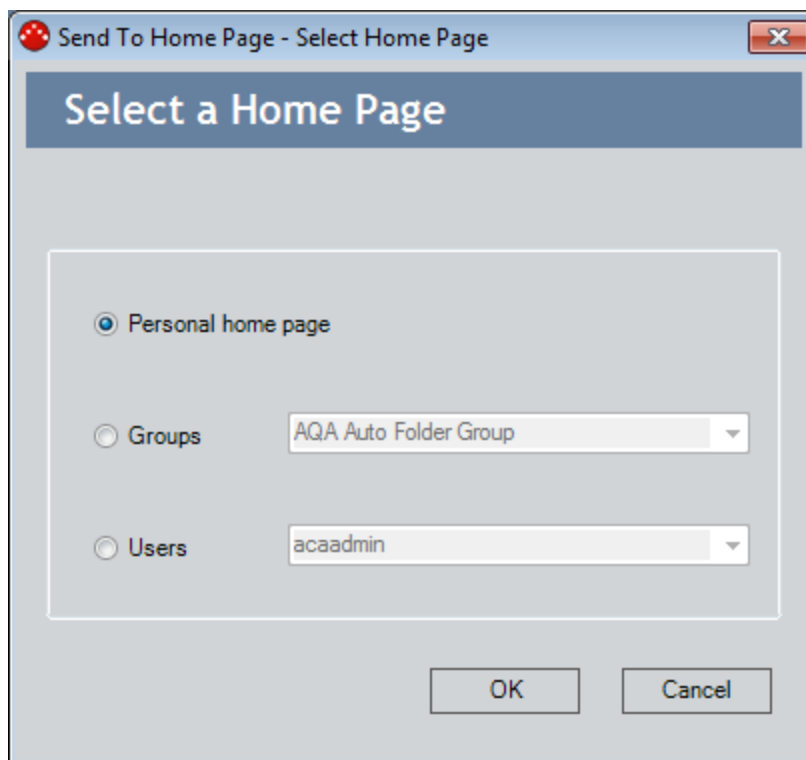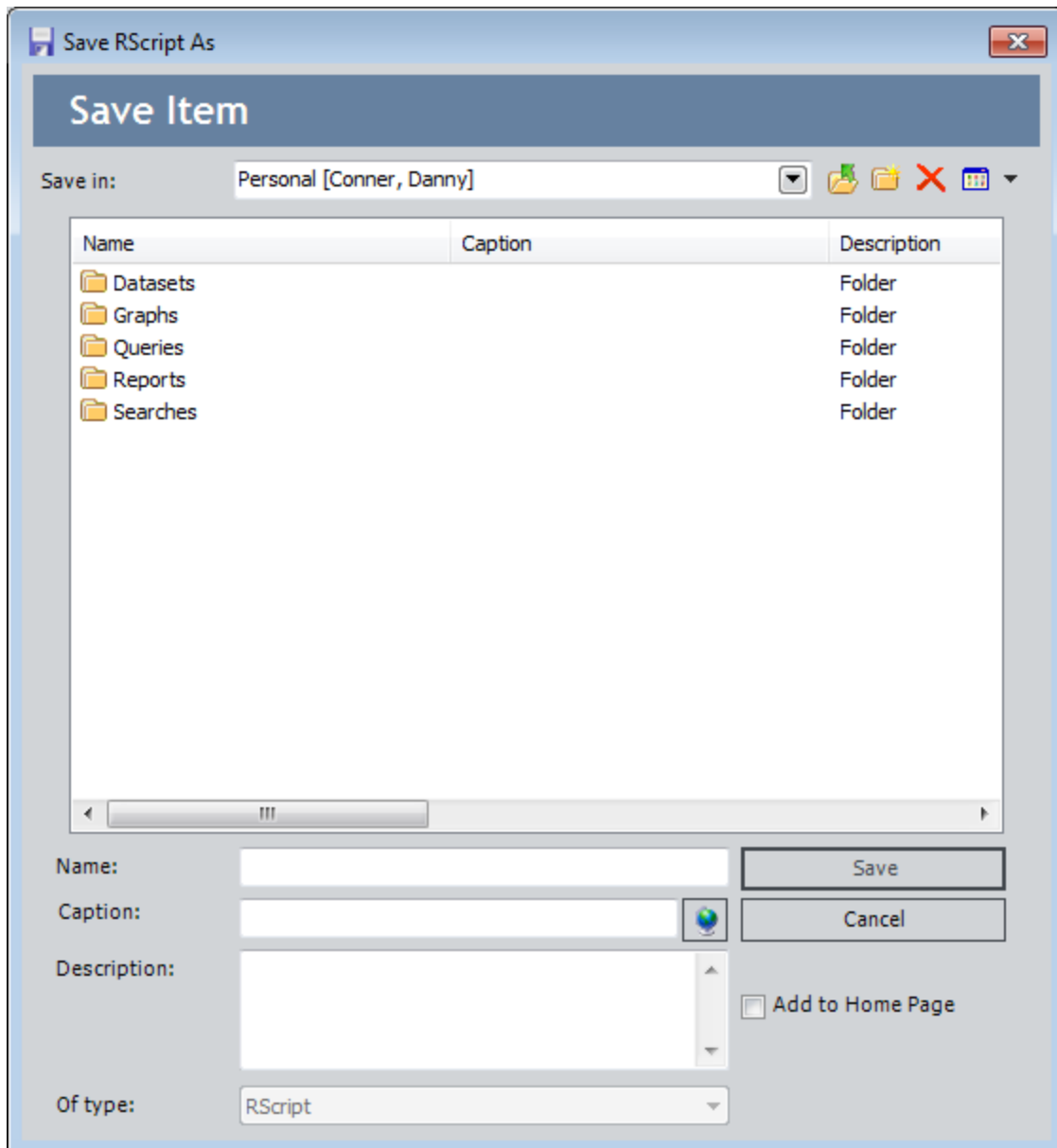
> **Note:** If you are viewing a saved R Script in the R Script Editor and you click the **Save As** link on the **Common Tasks** menu, the **Save R Script As** dialog box will appear so that you can save a copy of the current R Script with a different name or to a different location. You can use the following instructions to complete that task, as the process is the same as saving an R Script for the first time.

2. Navigate to the Catalog folder in which you want to save the R Script.

3. In the **Name** text box, type a name for the R Script. The name is required and must be unique to the Catalog folder in which you are saving the R Script.

4. The **Caption** text box is populated automatically with the value that you typed in the **Name** text box.

5. If desired, click the ![globe] button to manage translations for that string.

6. If desired, type a description of the R Script in the **Description** text box. This is not required to save the R Script.

7. Click the **Save** button.

   The R Script is saved to the Catalog.

Deleting R Scripts

To Delete an R Script:

1. While viewing the R Script in the **R Script Editor**, on the **Common Tasks** menu, click the **Delete R Script** link.

   The R Script is deleted. On the **Common Tasks** menu, the **Save R Script**, **Save R Script As**, and **Delete R Script** links are disabled.

   > **Note:** To re-enable the Save R Script, **Save R Script As,** and Delete R Script links, click New R Script, or open an R Script.

Options for Creating a New Strategy Record

You can create a new Strategy record in two ways:

- [You can create a new Strategy record from scratch](#).

  -or-

- [You can copy an existing Strategy record and use it as a template for creating a new Strategy record](#). This method is useful if you need to create two or more different Strategy records that vary slightly. Note that copying a Strategy record also copies the strategy rules associated with that record.

To create a Strategy record, you must:

- Be a member of the MI SM Administrator Security Group.
- Have Insert privileges to the Strategy family.
- Have View permissions to the following Catalog folder: \\Public\Rules Library\Meridium\Strategy Rules.
- Have Create permissions to the following Catalog folder: \\Public\Rules Library\Client\Strategy Rules.

Creating a New Strategy Record from Scratch

To create a new Strategy record from scratch:

1. In the Meridium APM Framework, on the **Strategy Start** page, click the **Create Strategy** link.

   The **Entity Family** dialog box appears.

2. In the list of families, select a family with which you want to associate the Strategy record.

3. Click **OK**.

   The **Create Strategy** page appears. The family you selected appears in the **Family ID** text box and is read-only.

4. In the **ID** text box, type the ID of the Strategy record. The ID is required and must be unique.

5. In the **Description** text box, type a description of thestrategy rules that you will associated with this record. The description is required.

6. In the **Purpose** text box, type the purpose of the strategy rules that you will associated with this record.

7. In the **Basis** text box, type the basis for the strategy rules that you will associated with this record.

8. In the **Engineering standard reference** text box, type information that references an industry standard (e.g., API [American Petroleum Institute] or ASME [American Society of Mechanical Engineers]) that applies to the strategy rules that you will associated with this record.

9. In the **Corporate guideline** text box, if your company has policies that should be reviewed or implemented when executing strategy rules that you will associated with this record, type a description of those policies.

10. In the **Owner** list, select the Security User who is responsible for creating and managing the strategy rules that you will associate with this record.

11. If the strategy rules that you will associate with this record will be sub-strategy rules, select the **Sub-Strategy** check box.

    > **Note:** The **Scheduled** check box is disabled. This property is set automatically when a Base Strategy is scheduled. Note also that the schedule for Sub-Strategies is defined by the Base Strategy that calls them.

12. If desired, select the **Spread** check box. If you select this check box, when *all* strategy rules for a family are executed for entities of a selected family, strategy rules that are associated with a family of the selected family will also be executed.

13. When you have finished defining all the properties for the Strategy record, on the

Current Strategy menu, click the **Save Strategy** link.

The Meridium APM system verifies that the Strategy ID is unique and that you have completed the required fields and saves the Strategy record to the database. If you did not enter all of the required information, an error message will be displayed. In addition, the **Click here to view Audit information** link appears at the bottom of the page.

> **Note:** After you have saved the new Strategy record, you can click the **Manage Rules** link on the **Common Tasks** menu to define rules associated with the Strategy record.

Creating a New Strategy Record by Copying an Existing Strategy

If you need to create two or more Strategy records that have similar properties but vary slightly, you can use an existing Strategy record as a template for creating the new Strategy record. Note that you cannot copy a record if the **Sub-strategy** check box is selected, and you can copy Strategy records only within the same associated family.

When you copy a Strategy record, any associated rules (base strategy rules or sub-strategy rules) are also copied.

To create a new Strategy record by copying an existing Strategy record:

1. In the Meridium APM Framework, on the **Strategy List** page, click the hyperlinked ID of the Strategy record that you want to use as the basis for the new Strategy record.

   The **Edit Strategy** page appears.

2. On the **Current Strategy** menu, click the **Copy Strategy** link.

   The **Edit Strategy** page reloads. The text **Copy n of** is added to the value in the **ID** text box, where **n** is a numeric value indicating the number of Strategy records that have already been created by copying this Strategy record. If this is the first copy you are creating, the text will be **Copy 0 of [Strategy ID]**, indicating that no Strategy records have been created from this Strategy record.

3. Modify the desired fields. In addition to the automatic modification the Meridium APM system made by inserting the text **Copy of** before the ID, you must modify at least one property before you can save the Strategy record.

4. When you are finished modifying the desired properties, on the **Current Strategy** menu, click the **Save Strategy** link.

   The Meridium APM system validates the properties and saves the new Strategy record.

Accessing a List of all Strategy Records

> **Note:** To access a list of all Strategy records, you must have at least View privileges to the Strategy family and be a member of the MI SM Administrator Security Group.

To access a list of all Strategy records:

- In the Meridium APM Framework, on the **Strategy Start Page**, click the **View Strategies** link.

  The **Strategy List** page appears, displaying a list of existing Strategy records. The list is grouped by the families to which the Strategy records are associated.

> **Note:** If you have accessed the **Strategy List** page before, the list may be filtered when you load it. In this case, all existing Strategy records and all Strategy record properties may not be displayed in the list. If the default properties are not displayed in the list, click the **Reset Strategy List** on the **Common Tasks** menu to reload the grid with the default options.

Aspects of the Strategy List Page

The list on the **Strategy List** page displays all the Strategy records that exist in the Meridium APM database. The list is grouped by the families with which the Strategy records are associated. For each Strategy record, the list includes the following columns of information:

- **ID:** The ID assigned to the Strategy record.
- **Description:** A brief description of the associated strategy rules.
- **Purpose:** The purpose of the associated strategy rules.
- **Engineering Standard Reference:** Information that references an industry standard (such as an API [American Petroleum Institute] or ASME [American Society of Mechanical Engineers]) that applies to the associated strategy rules.
- **Corporate Guideline:** Information about your company's policies that should be reviewed or implemented related to executing the associated strategy rules.

The **Strategy List** page contains the following task menu: **Common Tasks**.

Common Tasks Menu

The **Common Tasks** menu on the **Strategy List** page contains the following links:

- **Create Strategy:** Displays the Entity Family dialog box, where you can select the family with which you want to associate a new Strategy record.
- **Reset Strategy List:** Reloads the grid with the default options.
- **Execute Strategy(s):** Executes the strategy rules associated with the selected Strategy record. This link is enabled only after you select a row in the list.
- **Manage Rules:** Displays the **Meridium APM Rules Editor**, where you can manage the strategy rules associated with the selected Strategy record. This link is enabled only after you select a row in the list.
- **Delete Strategy(s):** Deletes the selected Strategy records from the database. This link is enabled only after you select a row in the list.
- **Send To:** Displays a submenu with options that let you .
- **Start Page:** Displays the **Strategy Start Page**.

Opening a Strategy Record

> **Note:** To open a Strategy record, you must at least View privileges to the Strategy family and be a member of the MI SM Administrator Security Group.

To open a Strategy record:

- In the Meridium APM Framework, on the **Strategy List** page, click the hyperlinked ID of the Strategy record that you want to view.

  The **Edit Strategy** page appears.

From the **Edit Strategy** page, you can:

- Modify the Strategy record.
- Manage the strategy rules associated with the Strategy.
- Execute the strategy rules associated with the Strategy record.
- Schedule the strategy rules associated with the Strategy record for execution.

Modifying a Strategy Record

The following instructions provide details on modifying a Strategy record.

> **Note:** To modify a Strategy record, you must have at least View privileges to the Strategy family and be a member of the MI SM Administrator Security Group.

To modify a Strategy record:

1. In the Meridium APM Framework, on the **Strategy List** page, click the hyperlinked ID of the Strategy record that you want to modify.

   The **Edit Strategy** page appears.

2. Make the desired changes. Note that you cannot modify the value in the Strategy ID cell or the Family ID cell.

3. If desired, click the **Click here to view Audit information** link to view historical information about the Strategy record, such as the Security User who last updated it and the date on which it was last updated. Note that you cannot modify this information.

4. When you are finished making changes, on the **Current Strategy** menu, click the **Save Strategy** link. Note that this link is enabled only after you have made at least one change.

   Your changes are saved to the database.

Deleting a Strategy Record

Note that if you delete a Strategy record whose rules reference a sub-strategy rule, the Strategy record associated with sub-strategy rule is not deleted.

> **Note:** You can delete custom Strategy records, but you cannot delete baseline Strategy records.

To delete one or more Strategy records:

1. In the Meridium APM Framework, on the **Strategy List** page, select the row(s) containing one or more Strategy records that you want to delete.

2. On the **Common Tasks** menu, click the **Delete Strategy(s)** link.

   A confirmation message appears, asking if you really want to delete the selected Strategy records.

3. Click the **Yes** button.

   The Strategy records are deleted from the database.

> Œ**Hint:** In addition to deleting Strategy records from the **Strategy List** page, you can delete a single Strategy record from the **Edit Strategy** screen. To do so, click the hyperlinked ID of the desired Strategy record on the **Strategy List** page. When the **Edit Strategy** screen appears, click the **Delete Strategy** link on the **Current Strategy** menu.

Accessing Historical Data About Strategy Records

To access historical data about Strategy record:

- In the Meridium APM Framework, on the **Edit Strategy** page, click the **Click here to view Audit information** link.

  The **Strategy Properties dialog box** appears.
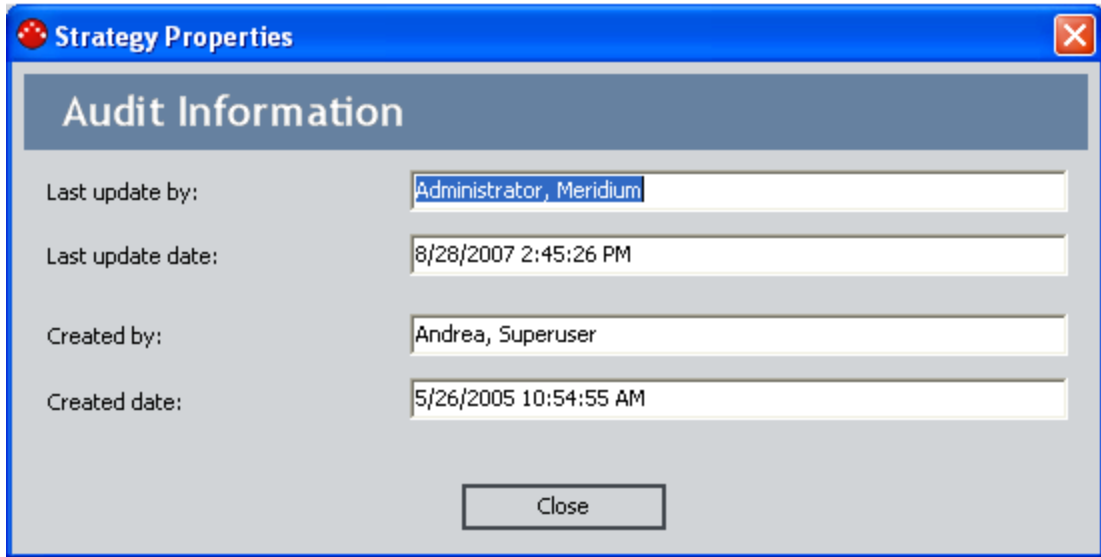
Aspects of the Strategy Properties Dialog Box

The **Strategy Properties** dialog box contains the following information:

- **Last update by:** The Security User who last updated the Strategy record.
- **Last update date:** The date on which the Strategy record was last updated.
- **Created by:** The Security User who created the Strategy record.
- **Created date:** The date on which the Strategy record was created.

> **Note:** These properties are view-only and cannot be modified. They are updated automatically when changes are made to the Strategy record.

About Creating and Modifying Strategy Rules

After you create a Strategy record, you must write strategy rules that tell the Meridium APM system what to do when they are executed. Strategy rules are stored in the Meridium APM Rules Library. When you create a Strategy record, the Meridium APM system automatically creates a Rules Library project for that Strategy record and stores it in the folder \\Client\Strategy Rules of the Meridium APM Rules Library. The name of the Rules Library project that is created corresponds to the ID of the Strategy record.

To write rules for a Strategy record, you must open the Rules Library project. You will be able to open the project and modify the rules only if VSTA has been installed on your computer and configured for use with Meridium APM. Via the Meridium APM Framework application, you will be able to open, modify, and compile the rules associated with individual Strategy records. Other features and functions of the Rules Library will not be available to you. To work with strategy rules within the Rules Library itself, you will need to use the Configuration Manager application.

To create or modify the strategy rules associated with a Strategy record:

1. Create a new Strategy record or open an existing Strategy record for which you want to define rules.

2. On the **Edit Strategy** page, on the **Common Tasks** menu, click the **Manage Rules** link.

   The **Meridium APM Rules Editor** appears, displaying the rules that are currently defined for the selected Strategy record.
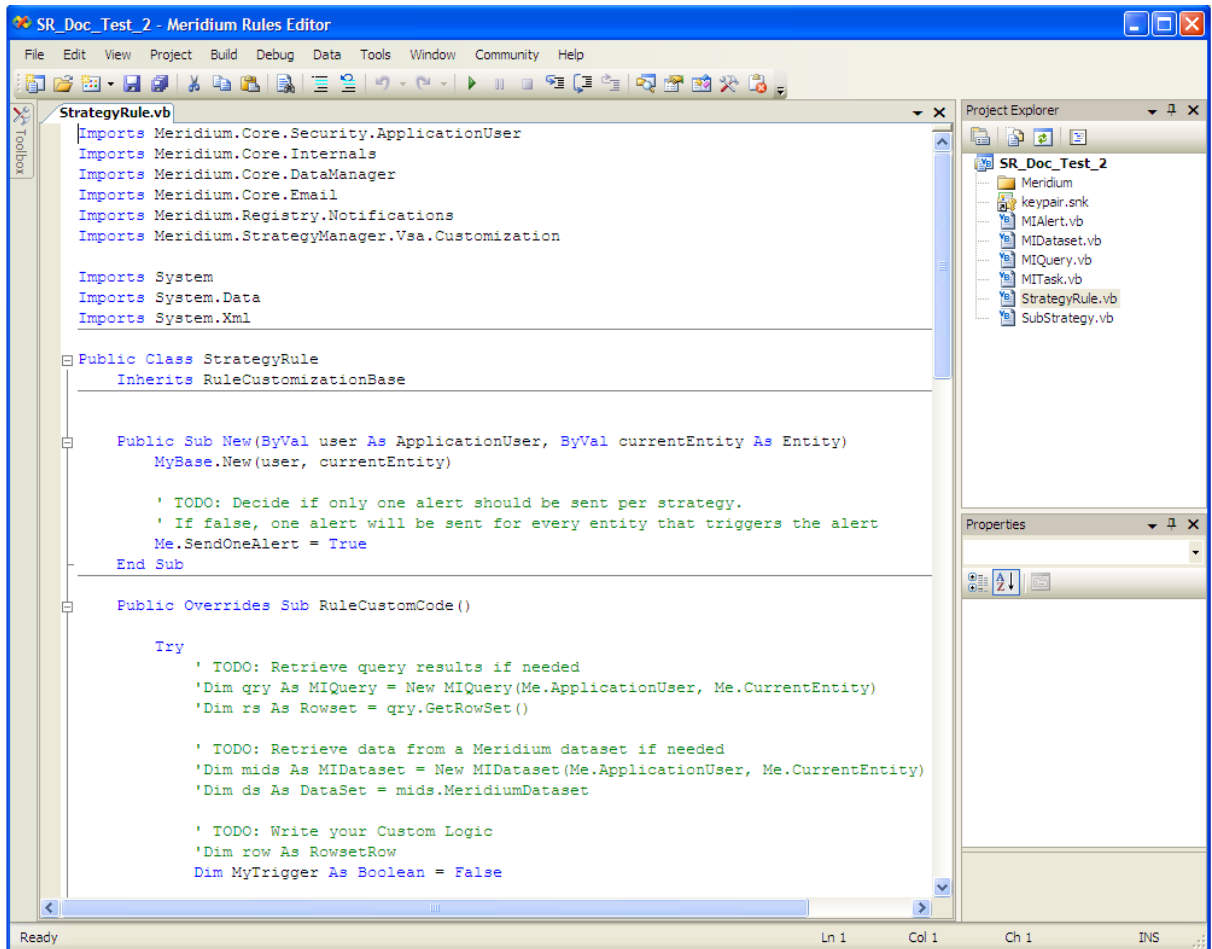
If you are managing rules that are associated with an *existing* Strategy record, the **Meridium APM Rules Editor** will display the rules that were last saved. If you are managing rules for a *new* Strategy record, the **Meridium APM Rules Editor** will display the rules defined in the Strategy Rule Template.

3. Edit the rule code as desired.

4. Compile and save the rules.

About the Meridium APM Strategy Rule Template

When you create a new Strategy record, the Meridium APM system automatically creates a Rules Library Project that stores the rules for that Strategy. The initial Rules Library project is created using a strategy rule template. The template is itself is stored in a Rules Library Project that exists for this purpose.

Meridium, Inc. provides a default strategy rule template via the Template project in the folder \\Meridium\Strategy Rules of the Meridium APM Rules Library. If you want to use your own template, you can create a project called **Template** in the folder *Client*\Strategy Rules of the Rules Library. If a Template project exists in the folder \\Client\Strategy Rules, it will be used. Otherwise, the default Meridium APM template will be used.

> **Note:** A Template project can be created in the folder \\Client\Strategy Rules via the Rules Library in the Configuration Manager application. The Template project cannot be created or managed via the Meridium APM Framework application. If you want to use the Meridium APM Template project as a starting point for creating your own Template project, open the **\\Meridium\Strategy Rules\Template** project in the **Meridium APM Rules Editor**, copy all the code, and paste it into a new ***Client*\Strategy Rules\Template** project. Then modify the project as desired.

The following information describes how to use and interpret the default Meridium APM template. This documentation explains only basic information for working with the template. It does not describe in detail the code that you can write to create rules. To interpret the template fully and write custom rules, you must have knowledge of VSTA and rule code.

If you create a new Strategy record and are using the default Meridium APM template, when you click the **Manage Rules** link on the **Common Tasks** menu, you will see a window similar to the one shown in the following image.
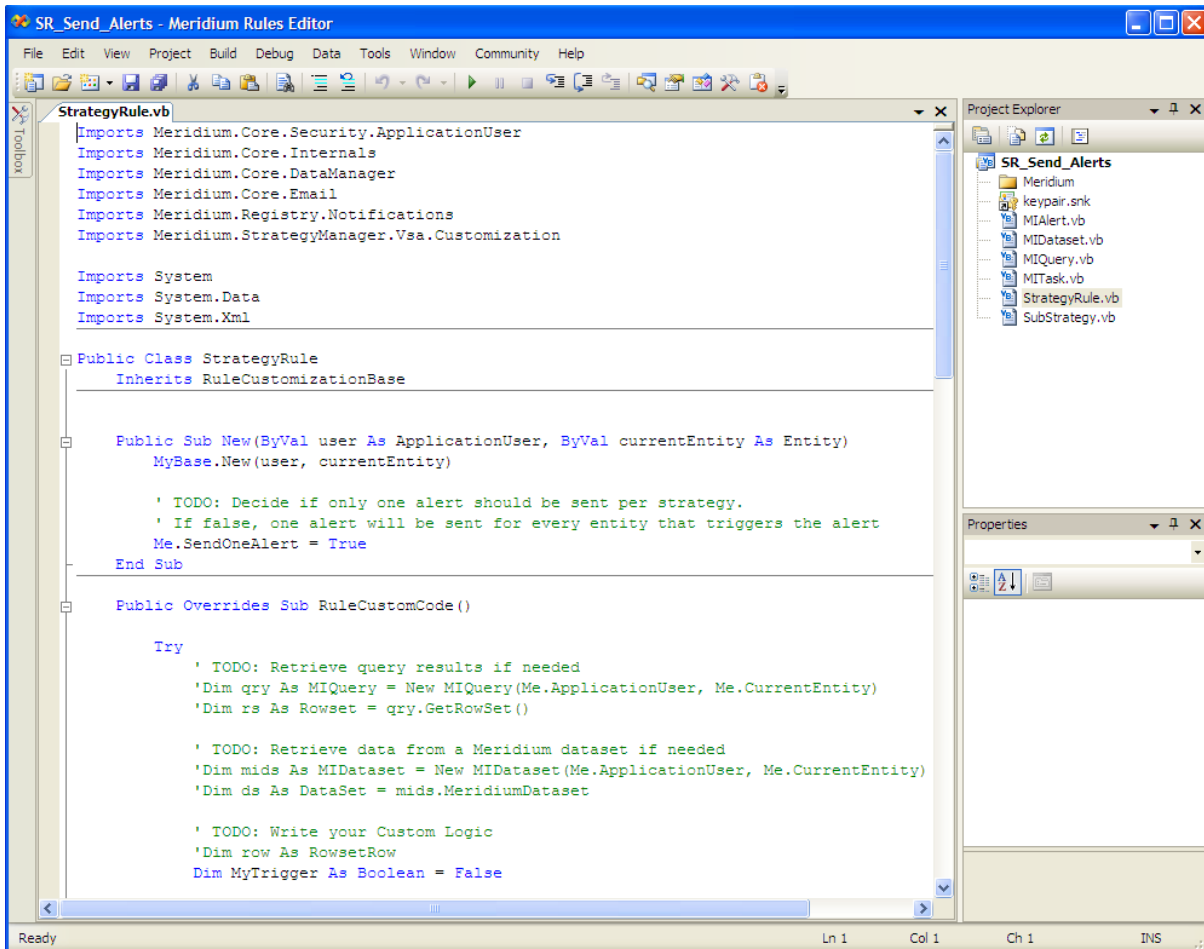
In the window header, you can see the text **SR_[Strategy ID] - Meridium APM Code Editor**, where **[Strategy ID]** is the ID of the Strategy record whose associated rules you are viewing. **SR_[Strategy ID]** is the name of the Rules Library project that was created to store the rules for the selected Strategy record.

As you scroll down the window, you will see a variety of **TODO** comments in green text. Each of these comments indicates something you might need to do to make the rules work. Note that you do not need to do everything indicated by a TODO comment. The comments provide a guideline for things you might want to do within the rule logic.

For example, suppose you want to send an email message to your night-shift inspector when a new pump is installed. To do so, your Meridium APM database might contain a query that returns all Pumps and their installation dates. When a new row is added to the results and the installation date is on or after the previous day, you will want your inspector to receive an email message so he can inspect the newly installed pump during his next shift.

To do so, in the rule logic, you would need to retrieve the query results, write the code that tells the Meridium APM system to do something when the query results meet a certain criteria, and set up the email message specifications.

You could use the first TODO comment as a reminder to retrieve the query results:

TODO: Retrieve query results if needed
        'Dim qry As MIQuery = New MIQuery(Me.ApplicationUser, Me.CurrentEntity)
        'Dim rs As Rowset = qry.GetRowSet()

You could then use the **TODO: Write your Custom Logic** comment as a reminder to write the code that tells the Meridium APM system which query results you are looking for before sending an email message to your inspector.

In addition, you could use the following comment as a guideline for writing code that will generate the email alert.

TODO: Create Alerts using the following code

Finally, you could use the **TODO: If sending alerts, provide information for the alert** comment to remember to complete the rule logic by specifying email message criteria, such as the email recipient and the subject line.

If desired, you can delete any rule code that does not apply to the specific rules you are defining.

About Executing Strategy Rules

When you execute strategy rules, the Meridium APM system performs the action identified by the rule logic. For example, the Meridium APM system might create one or more Alert records, create one or more Proposed Task records, or execute sub-strategy rules. Note that before you can execute a strategy rule, you must create the rules manually.

Strategy rules can be executed in two ways:

- You can execute strategy rules manually.
- The Meridium APM system can execute strategy rules automatically, according to a schedule defined in the Schedule Manager application.

Note: Before you can execute strategy rules, you must compile the strategy rule code. Otherwise, an error will appear when you try to execute the strategy rules.

Manually Executing Strategy Rules

To manually execute strategy rules:

1. In the Meridium APM Framework, on the **Strategy List** page, select one or more Strategy records whose strategy rules you want to execute.

2. On the **Common Tasks** menu, click the **Execute Strategy(s)** link.

   The **Rule Execution Progress** bar appears, indicating the progress of the strategy rule execution.

   The strategy rules are executed.

Scheduling Strategy Rules to Be Executed Automatically

You can schedule strategy rules to be executed automatically according to the schedule you specify. The scheduled item can then be viewed via the Schedule Manager application. Note that you can schedule only base strategy rules. Sub-strategy rules cannot be scheduled directly but will be executed on the same schedule that you define for the base strategy rules that call the sub-strategy rules.

To schedule strategy rules to be executed automatically:

1. In the Meridium APM Framework, on the **Strategy List** page, click the hyperlinked ID of the Strategy record whose strategy rule execution you want to schedule.

   The **Edit Strategy** page appears, displaying the Strategy record.

2. On the **Current Strategy** menu, click the **Schedule Strategy** link.

   The **Schedule** window appears, displaying any schedules that are currently defined.

   > **Note:** If a strategy rule is scheduled for execution but has not been executed yet, the Last Execution date will be 1/1/0001 12:00:00 AM.

3. To create a new schedule, click the **Add** link.

   > Œ Hint: You can also click the **Edit Selected** link to modify the selected schedule or the **Remove Selected** link to delete the selected schedule. Note that clicking the **Edit Selected** link will *clear the password* associated with the scheduled item. Even if you make no changes to the schedule itself, you will need to re-enter the password if you open an existing scheduled item.

   The **Schedule Item Properties** dialog box appears, displaying the **Schedule** tab by default.
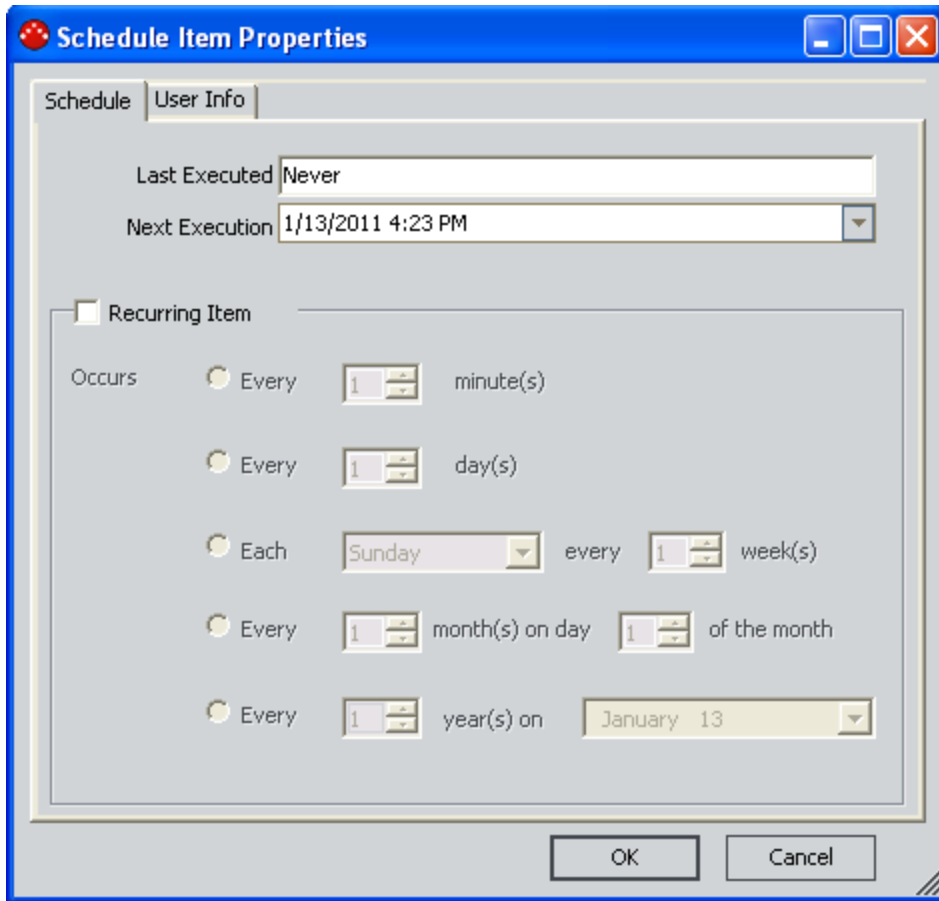
The **Last Executed Date** field displays the date on which the strategy rules were last executed. If the strategy rules have never been executed, **Never** appears in the **Last Executed Date** field.

4. If you want to schedule the strategy rules for one-time execution, in the **Next Execution** list, click the down arrow and select the desired date from the Calendar.

-or-

If you want the strategy rules to be executed more than once, select the **Recurring Item** check box, and then set the desired schedule:

- **Daily:** The strategy rules will be executed n number of days apart, as indicated by the value you select in the **Every __ days** list. For example, selecting 5 specifies that the strategy rules will be executed every 5 days. The default frequency is every day.

- **Weekly:** The strategy rules will be executed every n weeks, as indicated by the value that you select in the **every ___ weeks** list. For example, selecting 13 specifies that the strategy rules will be executed every 13 weeks. The strategy rules will be executed on the day of the week selected in the **Each** list. The default frequency is each Sunday of every week.

- **Monthly:** The strategy rules will be executed every n months, as indicated by the

value that you select in the first list. For example, if you select 3 in the first list, the strategy rules will be executed every third month. The value in the second list specifies the day of the month on which the strategy rules will be executed. The default frequency is every month on the first day of the month.

- **Yearly:** The strategy rules will be executed every n years, as indicated by the value that you select in the first list. For example, if you select 2 in the first list, the strategy rules will be executed every second year. The value in the second list specifies the month of the year and the day of the month on which the strategy rules should be executed. The default frequency is every year on the first day of the first month.

5. Click the **User Info** tab.

   The Meridium APM system automatically populates the fields on the **User Info** tab with your user information.

6. If you want to make the schedule available to a different user or a different data source instead of the current user or data source, you can enter another User ID and Password on this tab and select a different data source in the **Data Source** field.

7. Click **OK**.

   The Meridium APM system saves the schedule information and displays it in the list on the **Schedule** window.

8. Click the **Close** button to close the **Schedule** window.

About Task Records

Task records store information about a task that needs to be completed. Task records can represent recurring tasks or tasks that need to be completed only once.

Accessing the Task List Page

To access the Task List page:

- In the Meridium APM Framework, on the **Task Management Start page**, click the **Manage Tasks** link.

  The **Task List page** appears. By default, Task records for all Asset IDs and all of the Task subfamilies are displayed.

Aspects of the Task List Page

The **Task List** page lets you run the Task query to view a list of existing Task records that meet the query criteria.

The **Task List** page contains the following items:

- **Task menus:** Menus that provide specific functionality. The following task menus are available:
    - Selected Tasks
    - Asset Options
    - Optimize Tasks
    - Common Tasks
    - Associated Pages
- Task Management workspace: The area that allows you run the Task query to return a list of Task records that meet the query criteria.

From the **Task List** page, you can:

- Create a new Task record.
- Open an existing Task record.
- Schedule a Task record.
- Execute a Task record.
- Complete a recurring Task record.
- Propose changes to a Scheduled Task record.
- Reject a Task record.
- Access a list of Recommendation records that are linked to an Equipment or Functional Location record to which a Task record is linked.
- View revisions of all Task records that are linked to a given Equipment or Functional Location record.
- View revisions of a single Task record.
- View executions of all Task records that are linked to a given Equipment or Functional Location record.
- View executions of a single Task record.
- Refresh the query results.
- Access the Timeline.

Selected Tasks Menu
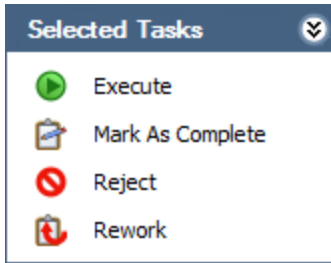
The **Selected Tasks** menu on the **Task List** page displays links that allow you to perform operations against a Task record or set of Task records. The links that are available depend upon the State Configuration that is defined for the Task family and the current state of the selected Task record(s).

For example, in the baseline State Configuration that is defined for the Task family, the state of a Task record can be changed from Scheduled to Rejected via the Reject operation. This means that if you select a Scheduled Task record in the grid on the **Task List** page, the **Selected Tasks** menu will contain the **Reject** link.

> **Note:** If you are not allowed to perform an operation on a Task record due to the limitations that are imposed by the State Configuration, when you select the row containing that Task record, the link corresponding to that operation will be disabled.

Asset Options Menu
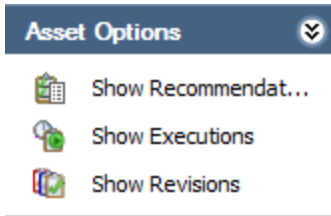
The **Asset Options** menu on the **Task List** page contains the following links:

- **Show Recommendations:** Displays the **Recommendation Management** page, which displays a list of Recommendation records that are linked to the Equipment or Functional Location record that is linked to the Task record that you selected on the **Task List** page.

  > Note: This link is enabled only if you select a Task record that is linked to an Equipment or Functional Location record that is also linked to a Recommendation record. In addition, note that if you select Task records that are linked to Equipment and Functional Location records, the **Recommendation Management** page will display all of the Recommendation records, grouped by Asset ID.

- **Show Executions:** Displays the [Task Execution History page](#), which displays executions of ALL Task records that are linked to the Equipment or Functional Location record to which the selected Task(s) record is linked.

  > Note: This link is enabled only if you select a Task record that is linked to an Equipment or Functional Location record that is linked to a Task Execution records.

- **Show Revisions:** Displays the [Task Revision History page](#), which displays revisions of ALL Task records that are linked to the Equipment or Functional Location record to which the selected Task(s) record is linked.

Optimize Tasks Menu
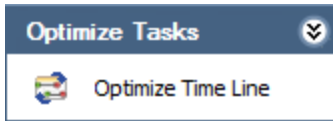
The **Optimize Tasks** menu on the **Task List** page contains the following link:

- **Optimize Time Line:** Displays the [Timeline](#) for the currently selected Task record (s).

  Note: This link is enabled only if a Scheduled Task record is selected in the grid.

Common Tasks Menu
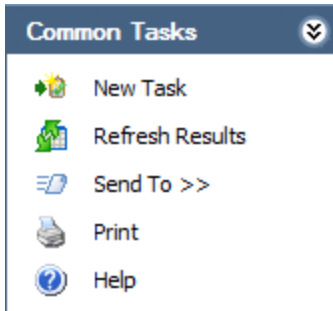
The **Common Tasks** menu on the **Task List** page contains the following links:

- **New Task:** Displays the **Task Builder**, where you can [create a new Task record](#).
- **Refresh Results:** [Reruns the Task query](#) using the current prompt values.
- **Send To:** Displays a submenu with options that let you .
- **Print:** Displays the Preview window, from which you can print the results in the grid.
- **Help:** Displays the context-sensitive Help topic for the **Task List** page.

Associated Pages Menu

The **Associated Pages** menu on the **Task List** page displays links that are configured for the Task family to which the displayed Task records belong.

Task Management Workspace

The **Task Management** workspace allows you run the Task query to return a list of Task records that meet the query criteria. The **Task Management** workspace is shown in the following image.

The baseline Tasks query is stored in the following Catalog folder: \\Public\Meridium\Modules\Task Management. This query contains the following prompts, which appear at the top of the page as lists:

- **Asset ID:** Displays a list of values in the Asset ID field in all existing Task records. Note that although Task Revision families are subfamilies of Task families, the query is configured to ignore Task Revision records.

  Œ Hint: On the default datasheet that is defined for all Task families, the Asset ID field is labeled Related Record ID.

- **Task Family:** Displays a list of all families that contain Task records with the selected Asset ID value.

Each time you access the **Task List** page, the values that appear by default in the prompts will be used to filter the list. After you select different criteria, you can click the

Run Query button to run the Task query and display the results in a grid at the bottom of the page.

The default Task query is configured to group the results by Asset ID. In addition, the default Task query is configured to display the following columns of information:

- **Current State:** The current state of the Task record.
- **Task ID:** The ID of the Task record. Clicking the hyperlinked ID will open the Task record in the Record Manager.
- **Asset ID:** The Record ID of the Equipment or Functional Location record to which the Task record is linked.
- **Task Description:** A description of the Task record.
- **Task Family:** The family to which the Task record belongs.
- **Last Date:** The last date on which the Task record was executed.
- **Next Date:** The next date on which the Task record should be executed. This is calculated based on the values in the **Last Date** and Desired Interval fields.
- **Desired Interval:** A number indicating how often the Task record should be executed.
- **Desired Interval UOM:** The unit of measure associated with the Desired Interval value.
- **Source Type**: The source of the Task record. For example, the source of the Task record may be the execution of a strategy rule.
- **Task Type:** The type of work that should be performed.
- **Reason:** The reason why the Task state was changed to Rejected, prepended with the user ID of the Security User who changed it.
- **Task Source ID:**The source of the Task record.

> **Note:** While all Task record fields are retrieved from the database when you run the query, all of the fields in this list may not be displayed on the **Task List** page if the display was previously customized. You can customize the list to display specific fields by right-clicking on a column header and clicking **Show Available Fields**.

Below the grid on the Task Management workspace, you will see a set of navigation options. These options are the same options that are displayed when you are viewing the query results.

About the Task Builder

The **Task Builder** is available in various places throughout the Meridium APM Framework application to facilitate the creation and modification of Task records. Besides Task Management, various other Meridium APM modules, such as Calibration Management, give you access to the **Task Builder**, which takes you step-by-step through the process of creating a Task record.

By using a URL, you can also make the **Task Builder** available anywhere in the Meridium APM Framework application that supports the use of links.

Using the Task Builder

The following instructions provide details on accessing the **Task Builder** from the **Task List** page. Note that these instructions include ALL steps presented in the **Task Builder**.

If you have accessed the **Task Builder** from a different location, your workflow may vary slightly. In these instructions, skip to the step that reflects the screen you see when you first launch the **Task Builder**. Additionally, you can skip over any step that shows a screen that you do not see while navigating through the **Task Builder**.

> **Note:** You cannot use the **Task Builder** to create Thickness Monitoring Task records. Thickness Monitoring can be customized to create Thickness Monitoring Task records automatically.

To create a new Task record using the Task Builder:

1. In the Meridium APM Framework, on the Task List page, on the **Common Tasks** menu, click the **New Task** link.

    The **Task Builder** appears, displaying the Welcome screen.
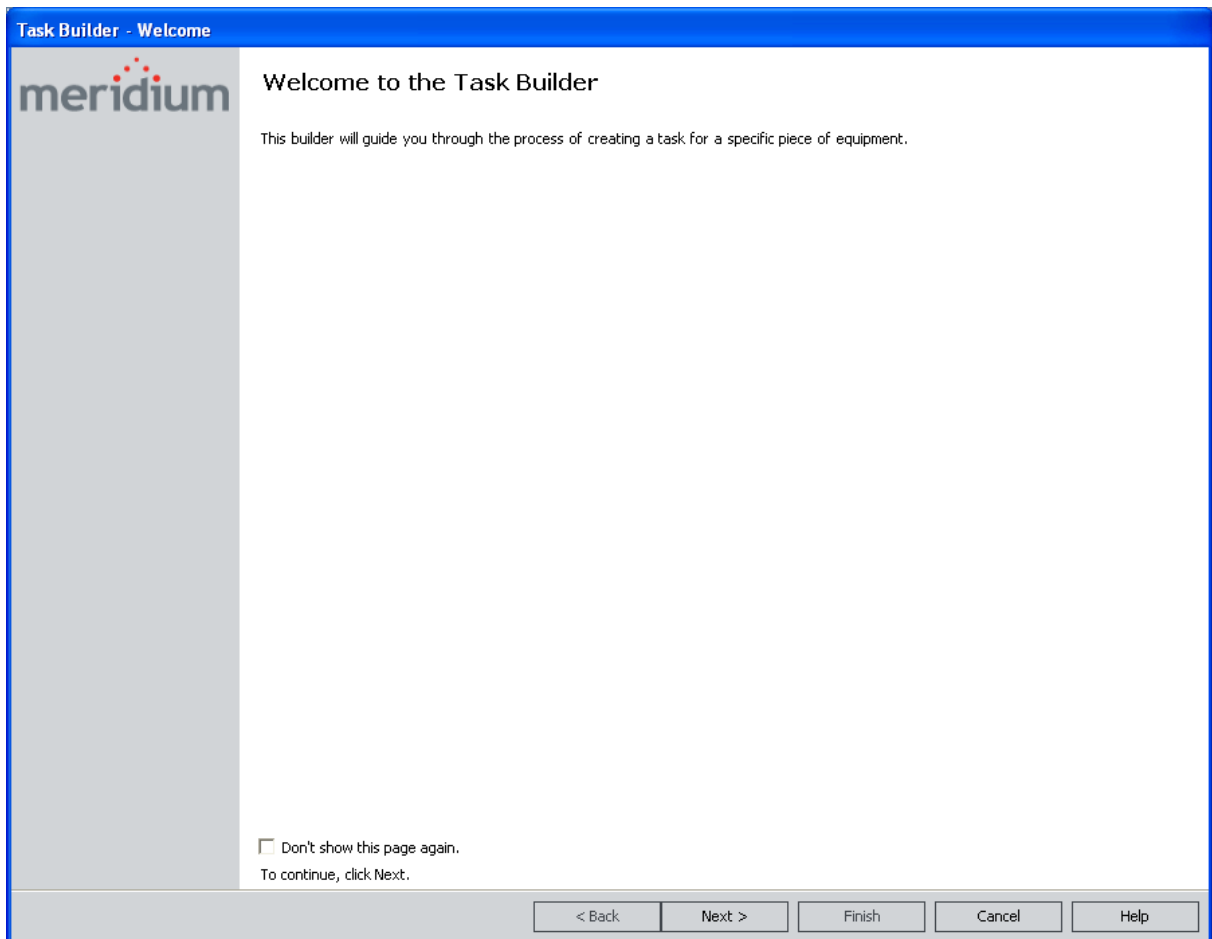
Note that if you do not want to view this screen in the future, you can click the **Don't show this page again** check box.

2. Click the **Next** button.

The **Equipment Selection** screen appears.

3. On the **Simple Search** tab, enter criteria to perform a Simple Search, or on the **Advanced Search** tab, enter criteria to perform an Advanced Search.

> **Note:** The **Search In** list displays families that are predecessors in the Has Tasks relationship.

4. Click the **Find Now** button to display a list of the records that meet your search criteria.

5. In the results, select the row containing the desired record, and click the **Next** button.

The **Task Family Selection** screen appears.
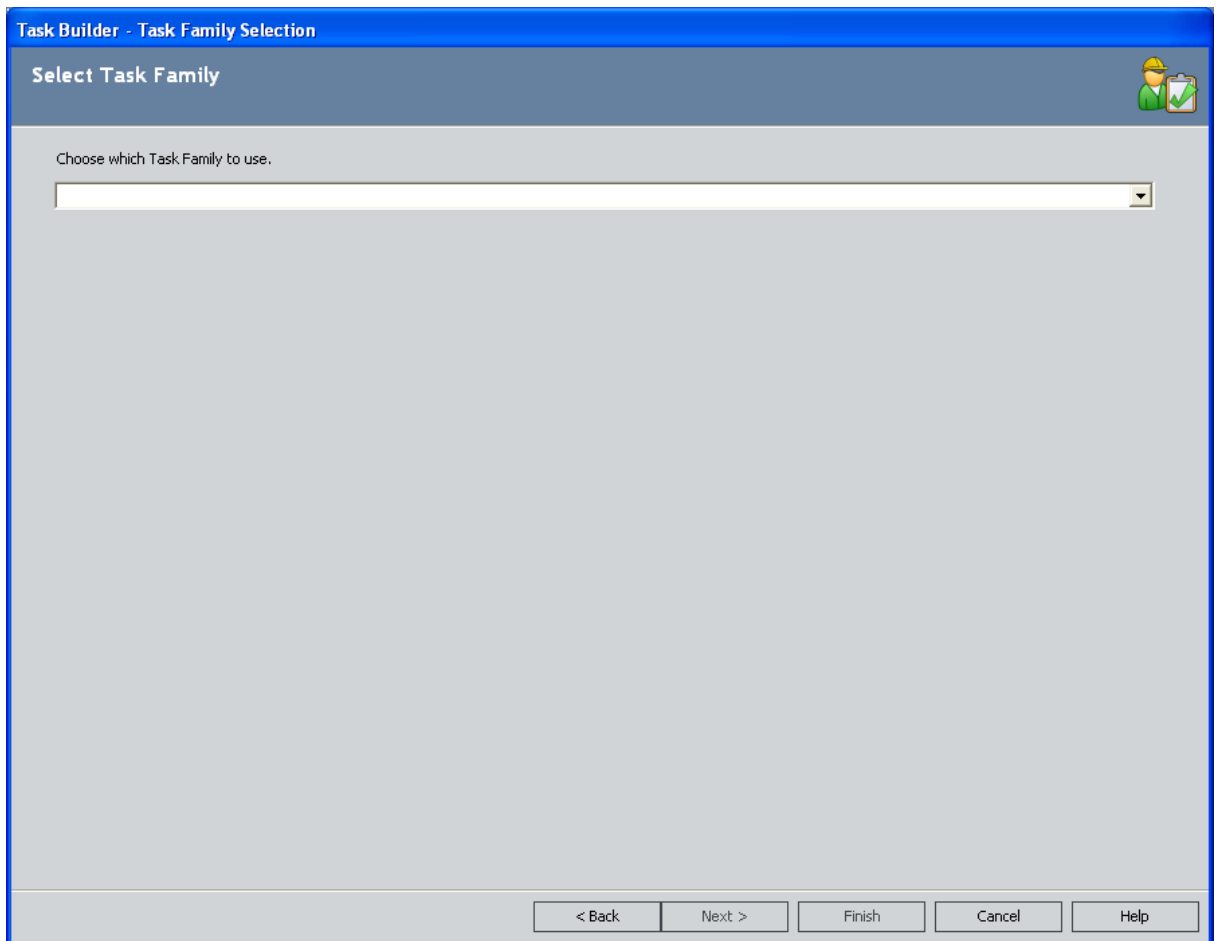
6. In the **Choose which Task Family to use** list, select the Task family that you want to use for creating the new Task record.

7. Click the **Next** button.

   The **New or Existing Task Selection** screen appears.
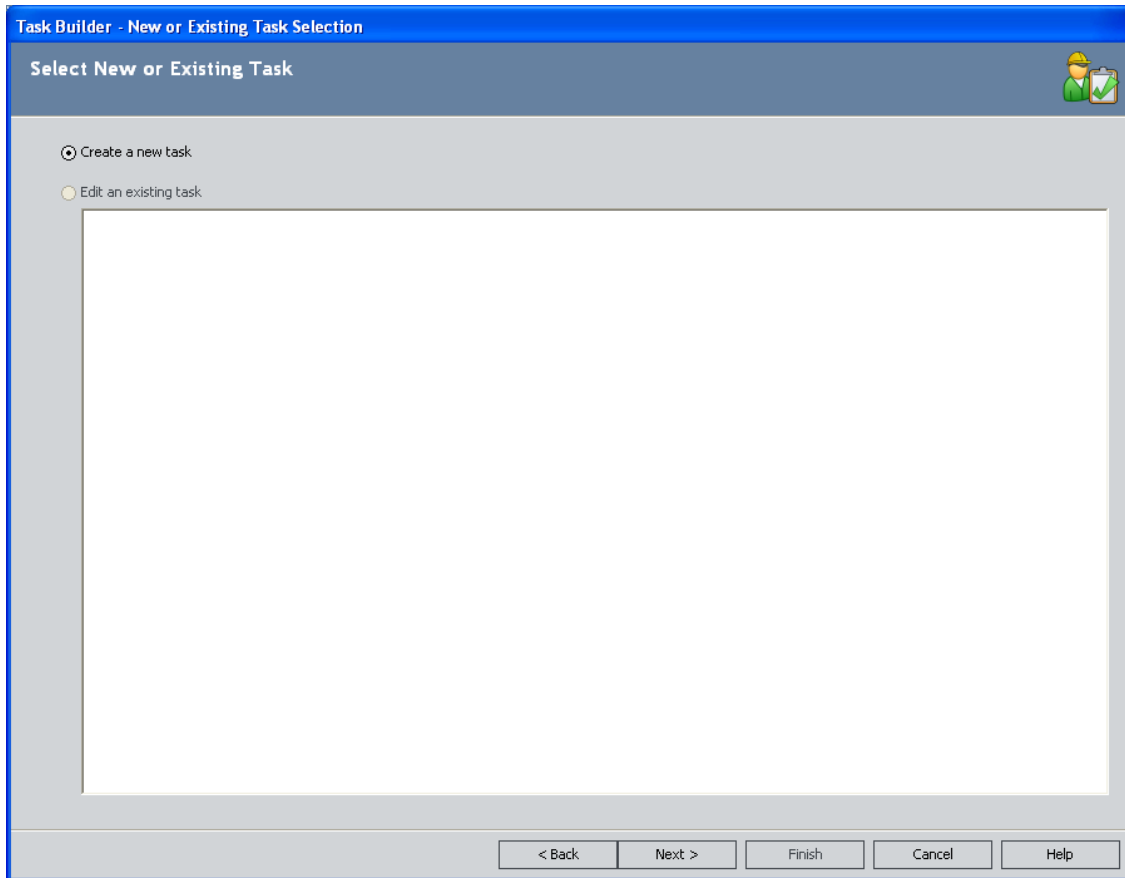
> **Note:** If Task records are already linked to the selected record, the Task records appear in a list on this screen. You can use the list to help you avoid creating duplicate Task records. If the desired Task record already exists, you can select the **Edit an existing task** option, select the Task record from the list, and then click the **Next** button to modify the existing Task record.

8. To create a new Task record, accept the default selection, **Create a new task**, and click the **Next** button.

   -or-

   To modify an existing Task record, select the **Edit an existing task** option, select the check box next to the Task record that you want to modify, and click the **Next** button.

   If you chose the **Create a new task** option on the previous screen, the **Create a new Task** screen appears, displaying a blank datasheet.

   -or-

   If you chose the **Edit an existing task** option on the previous screen, the **Edit a Task** screen appears, displaying the datasheet that was viewed when the Task record was last accessed.

9. Complete the available fields.

> **Note:** If you type a value in the **Desired Interval** cell, the value in the Maximum Interval field will be updated to mimic it. You can manually change the value in the **Maximum Interval** cell to be greater than the value in the **Desired Interval** cell. If, however, you reset the Desired Interval value to be greater than the Maximum Interval value, the Maximum Interval value will again be updated to mimic the Desired Interval value. In other words, the value in the Maximum Interval field must be equal to or greater than the value in the Desired Interval field.

10. Click the **Finish** button when you have completed the datasheet.

    The Task record is linked to the selected record.

Opening a Task Record

To open a Task record:

- In the Meridium APM Framework, on the **Task List** page, click the hyperlinked Task ID of the Task record that you want to view.

  The Task record appears in the Record Manager.

Scheduling a Task Record

To schedule a Task record:

1. In the Meridium APM Framework, access the **Task List** page.

2. Select prompt values that will cause the query to return the Task record(s) that you want to schedule, and click the **Run Query** button.

3. In the grid below the prompts, select the row(s) containing the Task record(s) that you want to schedule.

4. On the **Selected Tasks** menu, click the **Schedule** link.

   The Task state changes to Scheduled.

Executing a Task Record

To execute a Task record:

1. In the Meridium APM Framework, access the **Task List** page.

2. Select prompt values that will cause the query to return the Task record(s) that you want to execute, and click the **Run Query** button.

3. In the grid below the prompts, select the row(s) containing the Task record(s) that you want to execute.

4. On the **Selected Tasks** menu, click the **Execute** link.

   If the **Reoccurring** check box is not selected in the Task record, the state changes to Completed. The Completed state is a final state.

   -or

   If the **Reoccurring** check box *is* selected in the Task record, the state changes to Executing and then Scheduled.

   > Œ**Hint:** You will not see the Executing state in the interface. The Meridium APM system uses this state as a temporary state to transition the record back to the Scheduled state.

   From the Scheduled state, you can execute the Task record again whenever the task is performed again.

Completing a Recurring Task Record

If the Reoccurring check box is selected in a Task record, you can execute the Task record as many times as desired. If, however, you want to indicate the Task is complete and does not need to be executed again, you can set the state to Completed.

> **Note:** In the baseline State Configuration that is defined for the Task family, Completed is a final state. In other words, you cannot perform any other operations on the Task record after you complete it.

To complete a recurring Task record:

1. In the Meridium APM Framework, access the **Task List** page.

2. Select prompt values that will cause the query to return the recurring Task record(s) that you want to complete, and click the **Run Query** button.

3. In the grid below the prompts, select the row(s) containing the Task record(s) that you want to complete.

4. On the **Selected Tasks** menu, click the **Mark as Complete** link.

   The state changes to Complete.

Proposing Changes to a Scheduled Task Record

To propose changes to a Scheduled Task record:

1. In the Meridium APM Framework, access the **Task List** page.

2. Select prompt values that will cause the query to return the Scheduled Task record(s) to which you want to propose changes, and click the **Run Query** button.

3. In the grid below the prompts, select the row(s) containing the Task record(s) to which you want to propose changes.

4. On the **Selected Tasks** menu, click the **Rework** link.

   The state changes to Proposed.

Rejecting a Task Record

> **Note:** In the baseline State Configuration that is defined for the Task family, Rejected is a final state. In other words, you cannot perform any other operations on the Task record after you reject it.

To reject a Task record:

1. In the Meridium APM Framework, access the **Task List** page.

2. Select prompt values that will cause the query to return the Proposed or Scheduled Task records that you want to reject, and click the **Run Query** button.

3. In the grid below the prompts, select the rows containing the Task records that you want to reject.

4. On the **Selected Tasks menu**, click the **Reject** link.

   The **Task Management - Rejection Reason** dialog box appears.
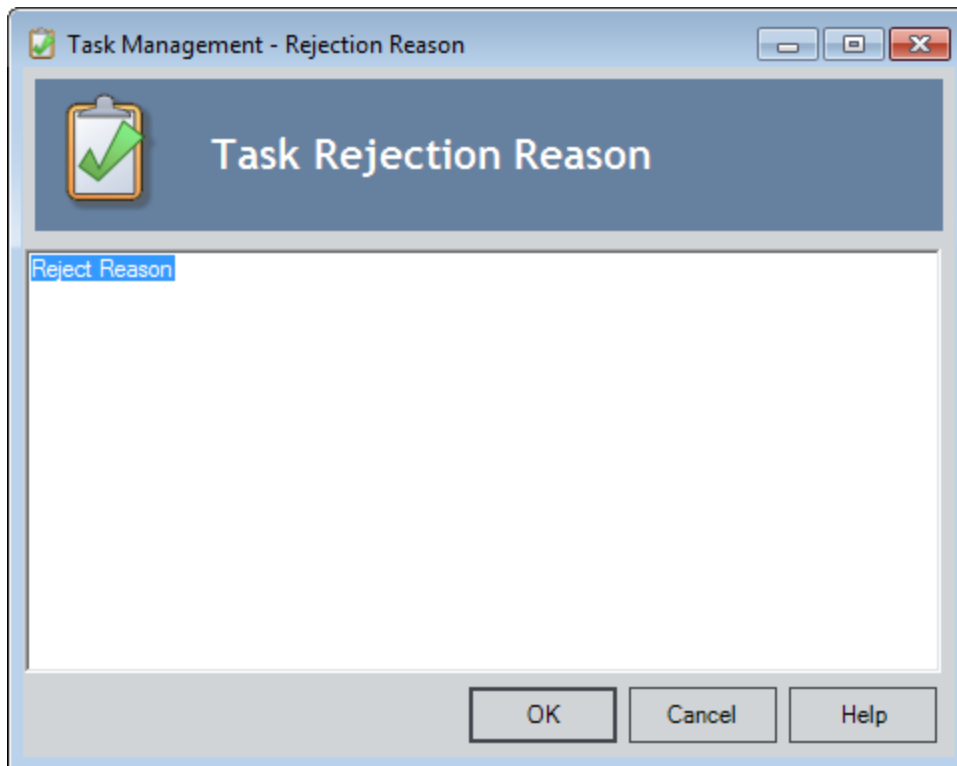
5. In the text area, delete the existing comment, and type comments indicating why you are rejecting the Task record.

6. Click **OK**.

If the **Rejectable** check box was selected in all of the selected Task records, the state of those Task records changes to Rejected. In addition, the **Reason** column displays the text that you typed, prepended with your user ID.

If, however, the **Rejectable** check box was selected in some of the selected Task records but not in others, a message appears, indicating that some of the Task records could not be rejected. For the Task records that were rejectable, the state changes to Rejected, and the **Reason** column displays the text that you typed, prepended with your user ID.

Accessing Recommendation Records

A Task record and a Recommendation record can be linked to the same Equipment or Functional Location record if the appropriate relationships are configured. For example, Task 1 can be linked to Equipment A if the Equipment family is related to the Task family through the Has Tasks relationship. Recommendation 1 can also be linked to Equipment A if the Equipment family is related to the Recommendation family.

If a Task record and a Recommendation record are linked to the same Equipment or Functional Location record, you can access that Recommendation record via the **Task List** page when you are viewing Task records that are linked to that Equipment or Functional Location record.

To access Recommendation records that are linked to an Equipment or Functional Location record to which a Task record is linked:

1. In the Meridium APM Framework, on the **Task List** page, select the row containing a Task record that is linked to the Equipment or Functional Location record whose linked Recommendation records you want to view.

2. On the **Asset Options** menu, click the **Show Recommendations** link.

   The **Recommendation Management** page appears, displaying a list of Recommendation records that are linked to the Equipment or Functional Location record that is linked to the selected Task record.

   > **Note:** If you selected rows containing Task records that are linked to different Equipment or Functional Location records, the **Recommendation Management** page displays Recommendations records that are linked to the Equipment or Functional Location records that are linked to each of the selected Task records.

Viewing Details of Associated Asset Strategies

In some cases, a Task record might have been used to create or update an Asset Strategy. In this case, the Task record will be linked to an Action record through the Has Action Driver relationship. In other cases, a Task record might have been created from an Action Revision record using Asset Strategy Implementation (ASI). In this case, the Task record will be linked to an Action Revision record through the Implements Action relationship.

If a Task record is linked to either an Action record through the Has Action Driver relationship or an Action Revision record through the Implements Action relationship, you can view the details of the associated Asset Strategy via the **Task List** page.

To view the details of the associated Asset Strategy:

1. In the Meridium APM Framework, on the **Task List** page, select the row containing a Task record that is associated with an Asset Strategy.

2. In the selected row, expand the plus sign.

   The **Strategy Details** tab appears.

   The following columns are displayed:

   • **Implementation Package:** Contains a hyperlink, which will open the **Packaging [Package ID]** page for the Implementation Package that contains the Task record.

   • **Asset Strategy:** Contains a hyperlink, which will open the **ASM - Asset Strategy Revision History - [Strategy ID]** page, where **[Strategy ID]** is the ID of the Asset Strategy that contains the Action record that was created from the Task record.

   • **Action:** Contains a hyperlink, which will open the **ASM - Asset Strategy Actions - [Strategy ID]** page. The Action record that was created from the Task record will be displayed in grid in the **Proposed Strategy** pane.

3. Click any hyperlink to display the associated information.

   The information is displayed.

Refreshing the Task Query Results

You can refresh the **Task List** page to rerun the Task query and display the most current information that exists in the database. For example, if any Task records have been added to the database and they meet the Task query criteria, these Task records will be displayed in the list. Note that when you refresh the list of Task records, the appearance of the grid is not reset. Any filters or sorting that you applied to the list will also continue to be applied.

To refresh the task list and obtain the latest data stored in the database:

- In the Meridium APM Framework, on the **Task List** page, on the **Common Tasks** menu, click the **Refresh Results** link.

  If any new Task records have been created or changes have been made to existing Task records since the last time you ran the Task query, the changes will be reflected in the grid.

About Task Execution Records

Task Execution records store information about when the work identified in a Task record was actually completed. A Task Execution record is created anytime you execute a Task record. When you execute a Task record, you are prompted for an execution date, which will be used to populate the Last Date field in the Task record and the Date Executed field in the linked Task Execution record.

Each Task Execution record is linked to the following records:

- The Task record that was executed.
- The Task Revision record that:
    - Is linked to the executed Task record.

      -and-

    - Contains the Next Date that is the closest to without exceeding the execution date.

In addition, Task Execution record can be linked to Inspection Event records if your database has been configured appropriately.

Note that when you execute a Task record and provide a date, the Task Execution record that is created will contain the provided date while the linked Task Revision record will contain the Last Date and Next Date that existed in the Task record *before* it was executed.

This approach provides useful information if the most current Task Revision record indicates that someone should execute a Task record on a certain date but that person does not actually execute the Task record until weeks later. You can compare the Next Date in the Task Revision record to the Date Executed in the Task Execution record to determine the difference between the suggested execution date (Next Date in Task Revision record) and the actual execution date (Date Executed in Task Execution record).

For example, suppose a Task Revision record contains the Next Date of 9/20/2008. If you do not perform the work and execute the linked Task record until 9/25/2008:

- The Date Executed in the Task Execution record will be 9/25/2008.
- The Next Date in the linked Task Revision record will still be 9/20/2008.

In other words, you can see from the Task Revision record that the work was supposed to be performed on 9/20/2008 but was not actually performed until 9/25/2008.

Note, however, that you can set the Date Executed in a Task Execution record to a date in the past. If you do so, the Task Execution record will be linked automatically to the Task Revision record with the Next Date this is closest to but does not exceed that execution date.

Consider the following situation. Suppose three Task Revision records exist for a single Task record, as shown in the following table.

Task Revision Number	Next Date
0	8/20/2008
1	9/20/2008
2	10/20/2008

You might perform the scheduled work on 9/22/2008 but forget to execute the corresponding Task record. Weeks later, on 10/25/2008, you remember to execute the Task record. When you do so, you select the execution date *9/22/2008* instead of the current date, 10/25/2008.

In this case, the Meridium APM system will locate the Task Revision record with the Next Date that is closest to but does not exceed the execution date of 9/22/2008. In this example, the Meridium APM system would locate *Task Revision 1* with the Next Date of *9/20/2008*. This Task Revision record would be linked automatically to the Task Execution record.

Illustration of Task Execution Records

Suppose you execute a Task record with the ID *Inspect Water Heater* and provide the execution date of 9/2/2008. In this case, the following records and links would exist.
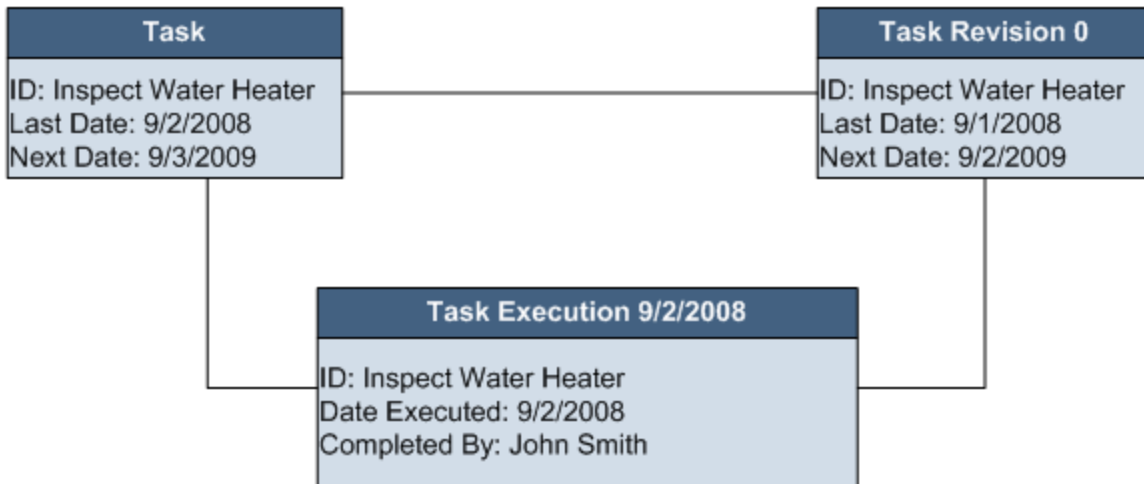
In the previous image, you can see that the Task Execution record is created and linked to both the Task record and the Task Revision record with the most current Revision Number (which is 0 in this case). The execution date of *9/2/2008* is used to populate the *Last Date* field in the Task record and the *Date Executed* field in the Task Execution record.

Note, however, that the Last Date and Next Date in the Task Revision record have not changed. This allows you to see the Last Date and suggested Next Date at the time that you executed the Task record.

If you were to execute the Task again on 9/3/2008, the following records and links would be created.

Here, the execution date of *9/3/2008* is used to populate the Last Date in the Task record and the Date Executed in the Task Execution record. The Last Date and Next Date in the Task Revision record, however, remain unchanged.

> **Note:** If the Last Date in a Task record has already been set to the execution date in one Task Execution record and you create a new Task Execution record with an *earlier* execution date, the Last Date will not be updated. The Last Date always reflects the *most current* execution date, which is not necessarily the *last entered* execution date. For example, suppose a Task record is already linked to a Task Execution record with an execution date of 11/15/2009, causing the Last Date in the Task record to be 11/15/2009. If you create a Task Execution record with an execution date of *10/15/2009*, the Last Date will not be updated.

Viewing Executions of ALL Task Records Linked to a Given Equipment or Functional Location Record

If an Equipment or Functional Location record is linked to multiple Task records, you can view those Task records and their linked Task Execution records.

To view executions of ALL Task records that are linked to a given Equipment or Functional Location record:

1. In the Meridium APM Framework, on the **Task List** page, select a row containing a Task record that is linked to the desired Equipment or Functional Location record.

2. On the **Asset Options** menu, click the **Show Executions** link.
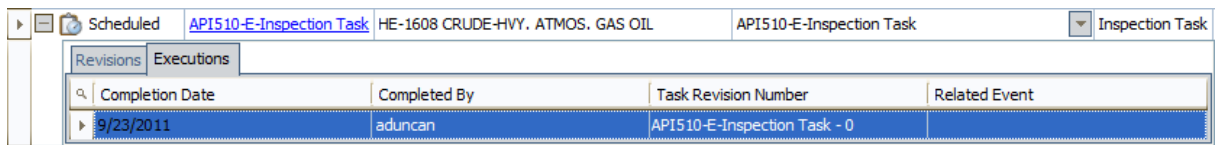
   The **Task Execution History** page appears.

Viewing Executions of a Single Task Record

To view all executions of a single Task record:

1. In the Meridium APM Framework, on the **Task List** page, select the row containing the Task record whose executions you want to view.

2. In the selected row, in the **Current State** column, click the plus sign.

    The **Executions** tab appears below the selected row. Each row on the **Executions** tab represents a Task Execution record that is linked to the selected Task record. The records are sorted in descending order according to the Date Executed. In the following image, you can see that only one Task Execution record is displayed in the grid.

3. To see the Task Revision record to which the Task Execution record is linked, click the hyperlink in the **Task Revision Number** cell.

    The Task Revision datasheet is displayed in a new window.

4. If the Task Execution record is linked to an Event record through the Has Event Execution relationship, to see the Event datasheet, click the hyperlink in the **Related Event** cell.

    The Event datasheet is displayed in a new window.

Aspects of the Task Execution History Page

The **Task Execution History** page contains the following items:

- **Common Tasks menu:** Provides access to common functionality.
- **Associated Pages menu:** Displays Associated Pages that have been configured for the family of the record that is selected in the **Task Execution Explorer** pane.
- **Task Execution Explorer pane:** Displays an Equipment or Functional Location record and its linked Task Execution records.
- **Task Execution Details pane** Displays the datasheet of the Task Execution record that is selected in the **Task Execution Explorer** pane.
- **Task Details pane** Displays the datasheet of the Task Revision record that is linked to the Task Execution record that is selected in the **Task Execution Explorer** pane.
- **Task Event Details pane** Displays the datasheet of the Event record that is linked to the Task Execution record that is selected in the **Task Execution Explorer** pane.
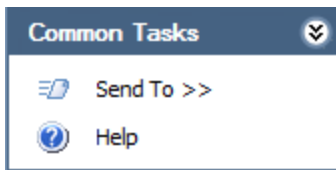
Common Tasks Menu

The **Common Tasks** menu on the **Task Execution History** page contains the following links:

- **Send To:** Displays a submenu with options that let you provide a link to the current page on your desktop (create shortcut), in an email message, or on a Home Page.

- **Print:** The **Print** link does not currently allow you to print the information that you see on the **Task Execution History** page.

  > Note: The **Print** link is enabled only if you select a Task Execution record in the **Task Execution Explorer** pane.

- **Help:** Displays the context-sensitive Help topic for the **Task Execution History** page.

Task Execution Explorer Pane

The **Task Execution Explorer** pane on the **Task Execution History** page contains a tree that displays an Equipment or Functional Location record and the Task Execution records that are linked to the Task record to which the Equipment or Functional Location record is linked. For example, suppose Equipment A is linked to Task 1, which is also linked to Task Execution 1. In this case, the **Task Execution Explorer** pane would display Equipment A and Task Execution 1.

The Equipment or Functional Location record that appears in the **Task Execution Explorer** pane is the Equipment or Functional Location record that is linked to the Task record that was selected on the **Task List** page from which you accessed the **Task Execution History** page.

The root level of the tree displays the Equipment or Functional Location record, which is represented by the ⚙ icon. The text describing the Equipment or Functional Location record is the Record ID of the Equipment or Functional Location record.

Directly below the Equipment or Functional Location record are nodes representing the execution date of the Task Execution records that are linked to a Task record to which the root Equipment or Functional Location record is linked. Each group is represented by the 📅 icon. The text describing the group is the execution date of the Task Execution records that are included in the group. Note that these groups are listed in the tree in descending order according to the date.

Below the date headings are nodes representing the Task Execution records that:

- Are linked to the Task record to which the root Equipment or Functional Location record is linked.
- Contain the Date Executed value that appears in the group header under which the records are listed.

Each Task Execution record is represented by the 🔖 icon. The text describing the Task Execution record is the Record ID of the Task Execution record.

Task Execution Details Pane

The **Task Execution Details** pane on the **Task Execution History** page displays the datasheet of the Task Execution record that is selected in the **Task Execution Explorer** pane.

The **Task Execution Details** pane is shown in the following image.
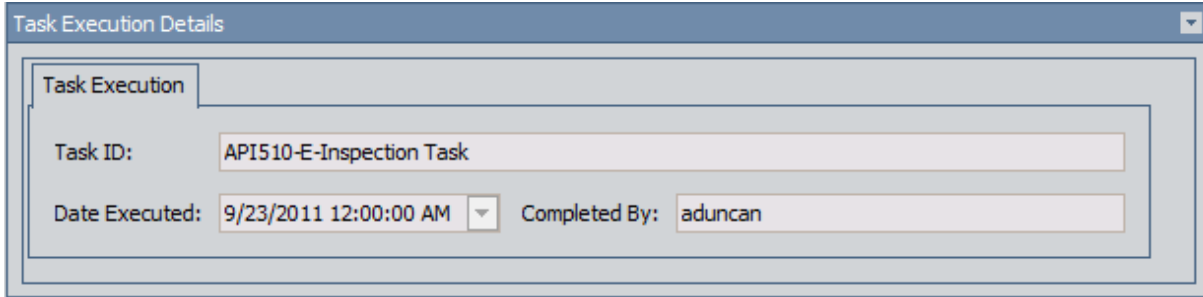
Note that all fields on the datasheet are disabled.

Task Details Pane

The **Task Details** pane on the **Task Execution History** page displays the datasheet of the Task Revision record that is linked to the Task Execution record that is selected in the **Task Execution Explorer** pane.

The **Task Details** pane is shown in the following image.
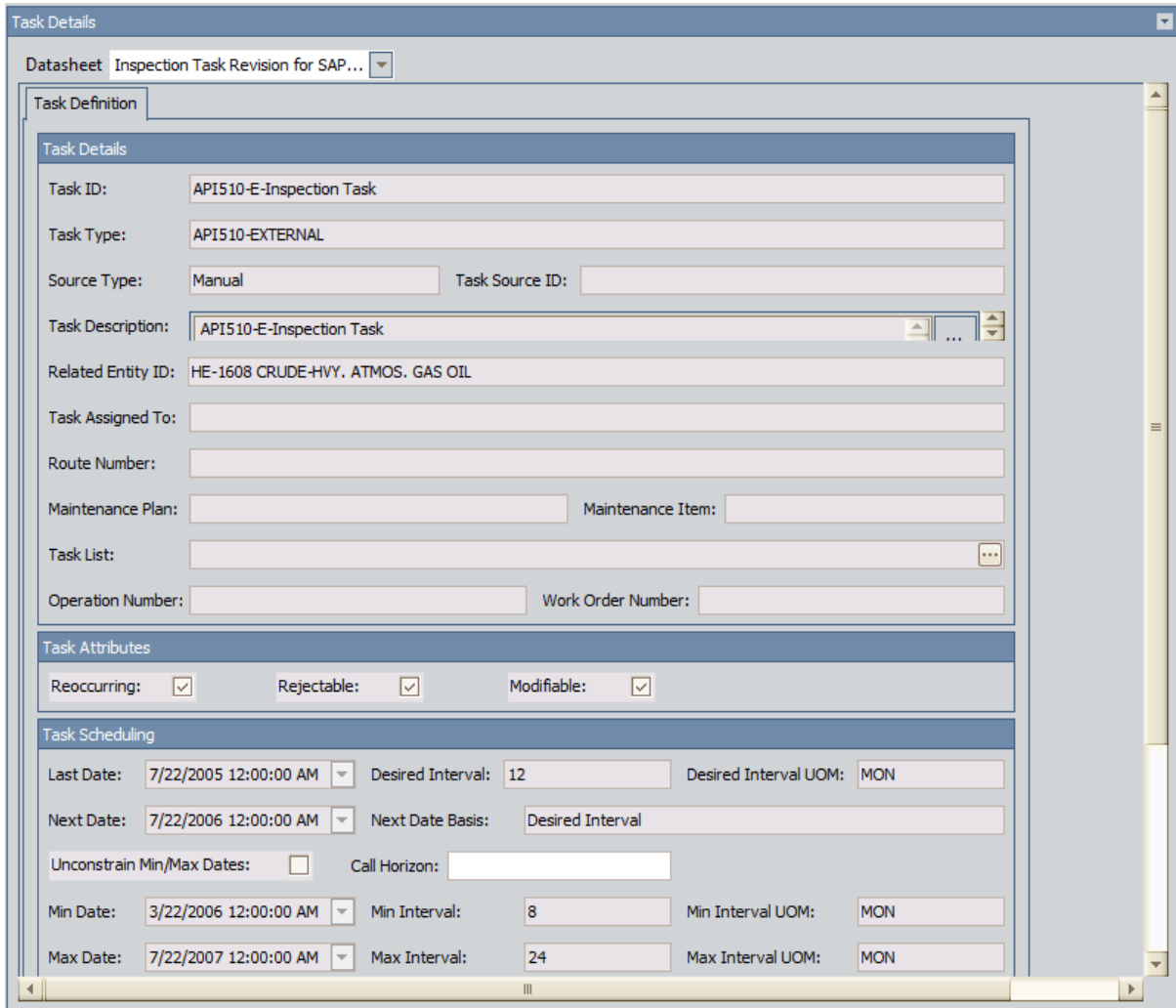
Note that all fields on the datasheet are disabled.

Task Event Details

The **Task Event Details** pane on the **Task Execution History** page displays the datasheet of the Event record that is linked to the Task Execution record that is selected in the **Task Execution Explorer** pane.

Note that all fields on the datasheet are disabled.

About Task Execution and Inspection Event Records

Using Inspection Management, you can create Inspection Event records that are linked to Inspection Task records. If certain conditions have been met, when you close the Inspection Event record, a Task Execution record will be created automatically and linked to the associated Inspection Task record.

Specifically, if an Inspection Task record is linked to an Inspection Event record, after you close and save the Inspection Event record, a Task Execution record will be created and linked to that Inspection Task record if the following conditions have been met:

- The Inspection Event family is related to the Task Execution family through the Has Event Execution relationship.
- The Inspection Task family is related to the Task Execution family through the Has Task Execution relationship.
- The Inspection Task record is set to a state of Scheduled.

> **Note:** To change the state of the Inspection Task record, State Configuration will need to be enabled for the Inspection Task family.

- The value in the Last Date field in the Inspection Task record is before the date on which the Inspection Event record was closed.

When the Task Execution record is created, the value in the Date Executed field will be set automatically to the date on which the Inspection Event record was closed.

Also, if the Inspection Task Revision family is related to the Task Execution family through the Has Task Execution relationship, the Task Execution record will also be linked to the Inspection Task Revision record that is:

- Linked to the Inspection Task record

  -and-

- That contains the Next Date that is the closest to without exceeding the execution date.

An Inspection Example

The following example illustrates how Task Execution records are created and linked automatically to Inspection Task records that are linked to Inspection Event records. Suppose that the following conditions are true:

- The following relationships exist.
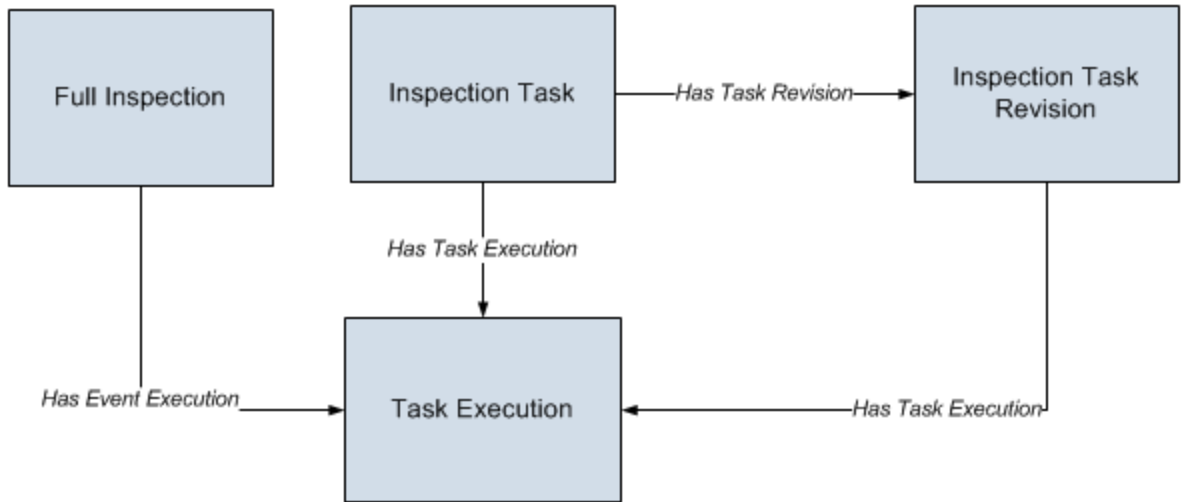
- The Inspection Task record *Inspection Task 1* is linked to the Pump record *Pump 101*.
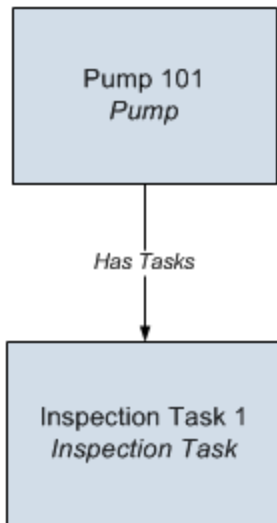
- The state of the Inspection Task record is *Scheduled*.
- The value in the Last Date field in the Inspection Task record is *2/17/2009*.
- The Inspection Task Revision record *Inspection Task 1 - Rev 0* is linked to the Inspection Task record *Inspection Task 1*.
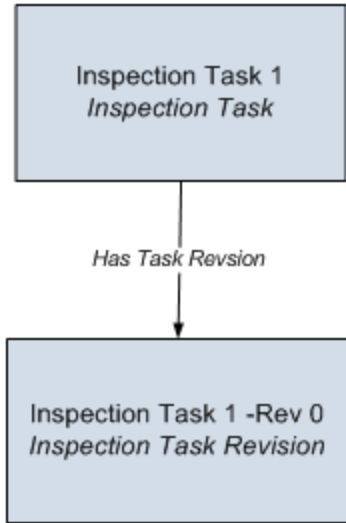
Now, assuming that the described conditions exist, suppose you create the Full Inspection record *Full Inspection 1* and link it to the record *Pump 101*, which is linked to the Inspection Task record *Inspection Task 1*.
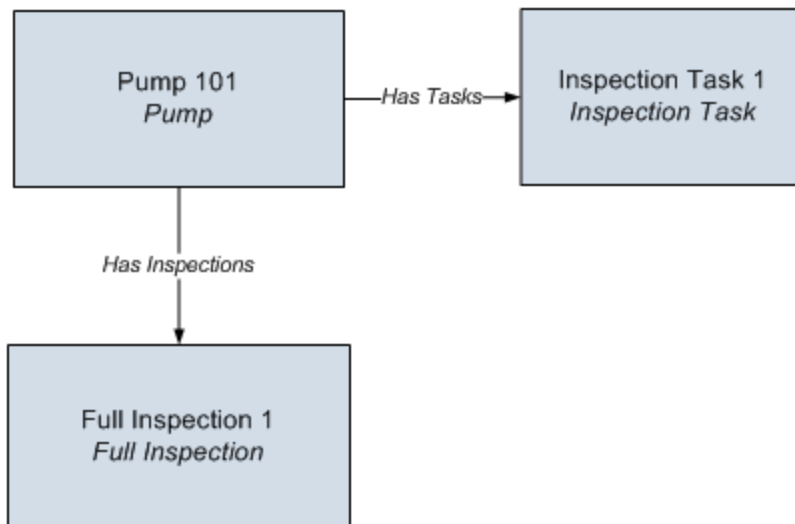
On 3/17/2009, you close the Full Inspection record.

In this case:

- A Task Execution record is created automatically and linked to the following records:
    - Full Inspection record *Full Inspection 1*.
    - Inspection Task record *Inspection Task 1*.
    - Inspection Task Revision record *Inspection Task 1 - Rev 0*.
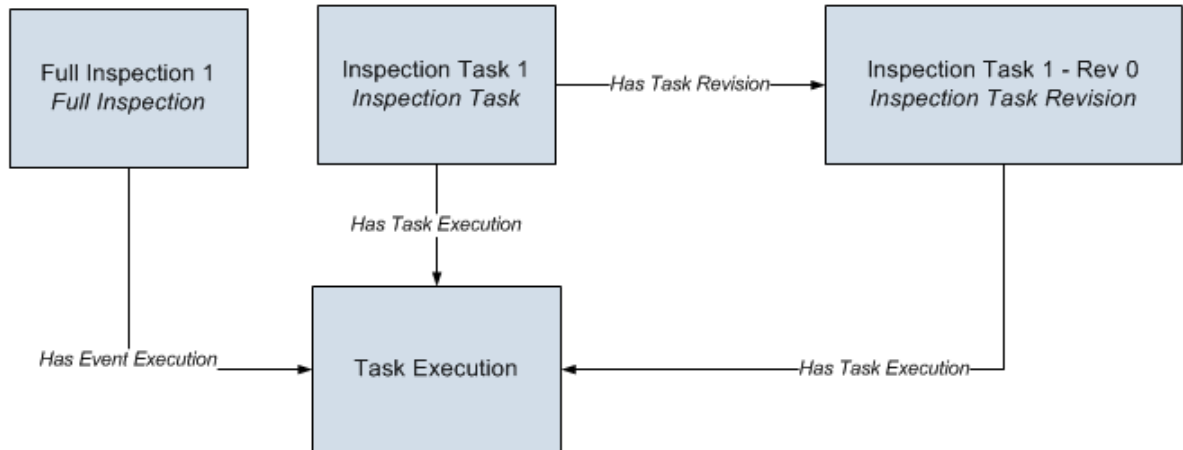
- The value in the Date Executed field in the Task Execution record is set automatically to *3/17/2009.*

About Task Execution and Calibration Event Records

Using Calibration Management, you can create Calibration Event records that are linked to Calibration Task records. If certain conditions have been met, when you close the Calibration Event record (via the **Calibration Closed** check box), a Task Execution record will be created automatically and linked to the associated Calibration Task record.

Specifically, if a Calibration Task record is linked to a Calibration Event record, after you select the **Calibration Closed** check box and save the Calibration Event record, a Task Execution record will be created and linked to that Calibration Task record if the following conditions have been met:

- The Calibration Event family is related to the Task Execution family through the Has Event Execution relationship.
- The Calibration Task family is related to the Task Execution family through the Has Task Execution relationship.
- The Calibration Task record is set to a state of Scheduled.

  > Note: To change the state of the Calibration Task record, State Configuration will need to be enabled for the Calibration Task family.

- The value in the Last Date field in the Calibration Task record is before the date that the **Calibration Closed** check box is selected.

When the Task Execution record is created, the value in the Date Executed field will be set automatically to the date that the **Calibration Closed** check box was selected.

Also, if the Calibration Task Revision family is related to the Task Execution family through the Has Task Execution relationship, the Task Execution record will also be linked to the Calibration Task Revision record that is:

- Linked to the Calibration Task record

  -and-

- That contains the Next Date that is the closest to without exceeding the execution date.

An Example

The following example illustrates how Task Execution records are created and linked automatically to Calibration Task records that are linked to Calibration Event records. Suppose that the following conditions are true:

- The following relationships exist.
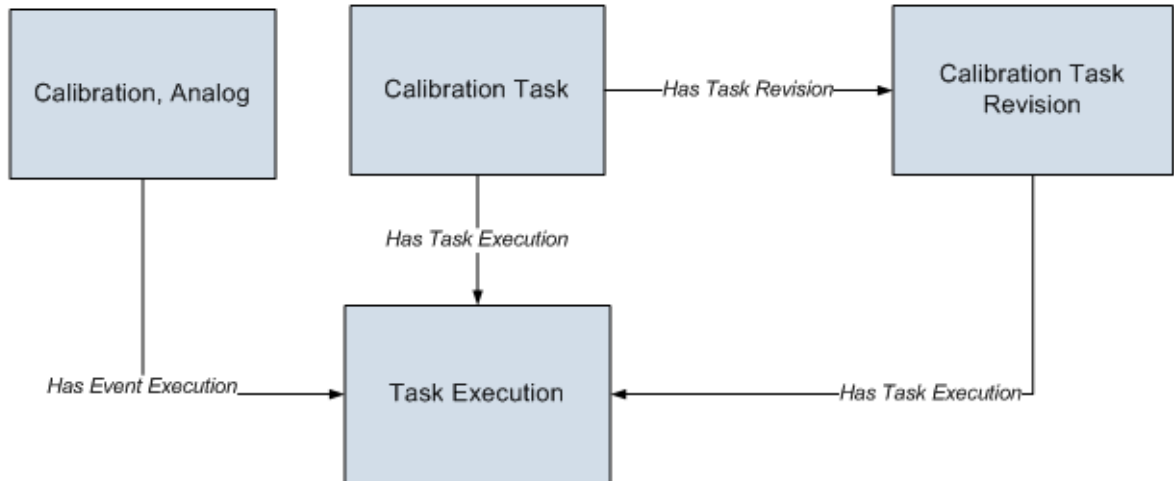
- The Calibration Task record *Calibration Task 1* is linked to the Pump record *Pump 101*.
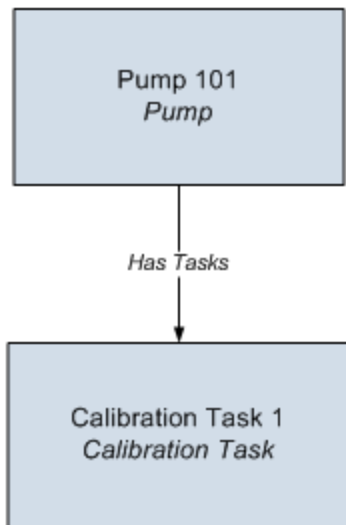
- The state of the Calibration Task record is *Scheduled*.
- The value in the Last Date field in the Calibration Task record is *2/17/2009*.
- The Calibration Task Revision record *Calibration Task 1 - Rev 0* is linked to the Calibration Task record *Calibration Task 1*.
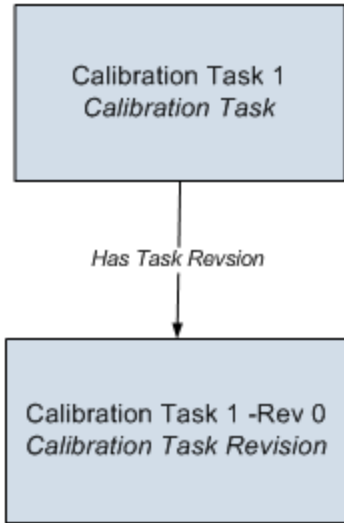
Now, assuming that the described conditions exist, suppose you create the Calibration, Analog record *Calibration, Analog 1* and link it to the record *Pump 101*, which is linked to the Calibration Task record *Calibration Task 1*.
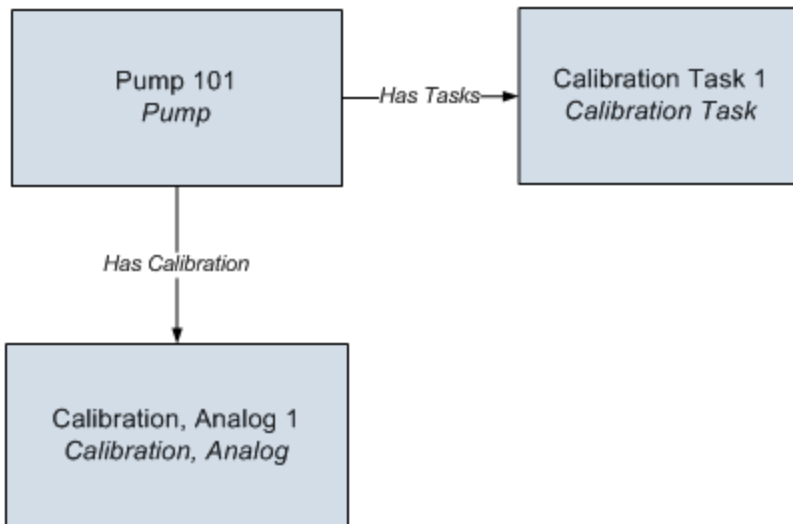
On 3/17/2009, you select the **Calibration Closed** check box in the Calibration, Analog record.

In this case:

- A Task Execution record is created automatically and linked to the following records:
  - Calibration, Analog record *Calibration, Analog 1*.
  - Calibration Task record *Calibration Task 1*.
  - Calibration Task Revision record *Calibration Task 1 - Rev 0*.
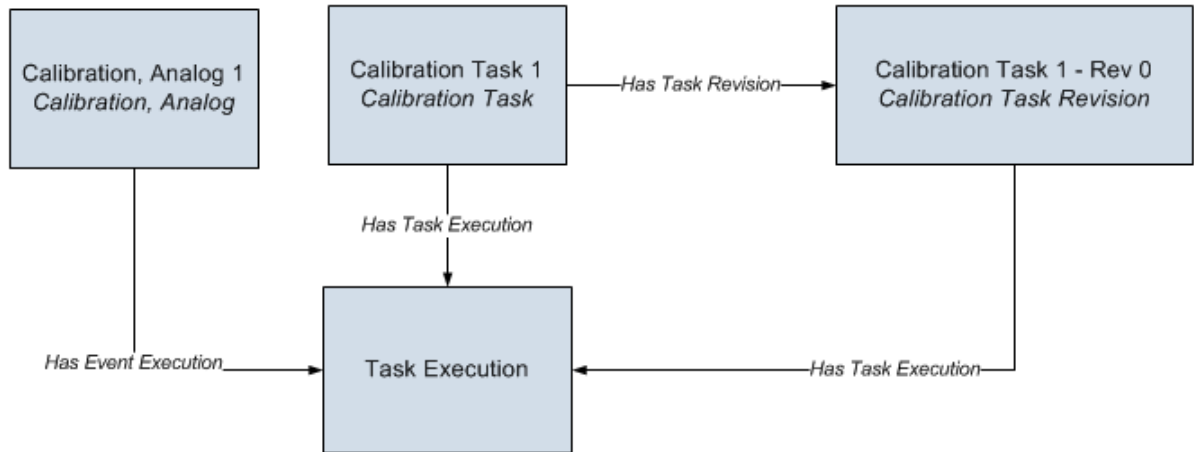
- The value in the Date Executed field in the Task Execution record is set auto-matically to *3/17/2009*.

About Task Revision Records

Task Revision records provide an historical record of the information in a Task record on a particular date and time. The baseline Meridium APM database contains the following Task Revision families:

- Task Revision
- Calibration Task Revision
- Inspection Task Revision

Note that throughout this documentation, in most cases, we use the term *Task Revision record* to refer generally to a record in any of these three Task Revision families.

If you have configured the Has Task Revision relationship correctly, when you create a Task, Calibration Task, or Inspection Task record, a Task Revision, Calibration Task Revision, or Inspection Task Revision record is created and linked automatically. The field values from the Task record are copied to the Task Revision record.

If you later modify a value in a Task record, a new Task Revision record is created and linked to the Task record, and the updated Task record values are copied to the new Task Revision record. While the Task record always contains current information, Task Revision records allow you to see how the values have changed over time.

Note that changes to all Task record values *except* those in the following list will result in the creation of a new Task Revision record:

- [MI_SM_STATE_ID_C]
- [MI_SM_STATE_ENTERED_D]
- [MI_SM_STATE_OWNER_ID_C]

> **Note:** The fields listed above are State Configuration fields, which are populated automatically when you perform an operation on a Task record. In other words, when you perform an operation on a Task record (e.g., schedule a Task record), a Task Revision record is not created.

- Last Date
- Next Date
- Min Date
- Max Date
- Next Date Basis (Inspection Task records only)

> **Note:** Changing the value in the Next Date Basis field in an Inspection Task record will not cause the Meridium APM system to create a new Inspection Task Revision record. Changing the value in the Next Date Basis field in a Task record or Calibration Task record, however, *will* cause the Meridium APM system to create a new Task Revision or Calibration Task Revision record.

Each Task Revision record has a Revision Number field, which stores a numeric value indicating the number of Task Revision records that are linked to the Task record. Note that the *first* Task Revision record that is created and linked to a Task record will have a Revision Number of 0. The second Task Revision record that is created and linked to the same Task record will have a Revision Number of 1, and so on.

For example, suppose Task Record A is written today. When Task Record A is saved, Task Revision Record 0 will be created and linked to Task Record A. If tomorrow you were to modify the Task ID in Task Record A, Task Revision Record 1 would be created and linked to Task Record A. If, however, you were to modify the Last Date in Task Record A the next day, a new Task Revision record would not be created. Instead, a Task Execution record would be created.

Illustration of Task Revision Records

Suppose you create a Task record to inspect the water heater in your home. In this case, the following records would be created and linked.

Task

ID: Inspect Water Heater
Description: Full Inspection of Water Heater
Last Date: 9/5/2008
Next Date: 9/5/2009

Task Revision 0

ID: Inspect Water Heater
Description: Full Inspection of Water Heater
Last Date: 9/5/2008
Next Date: 9/5/2009
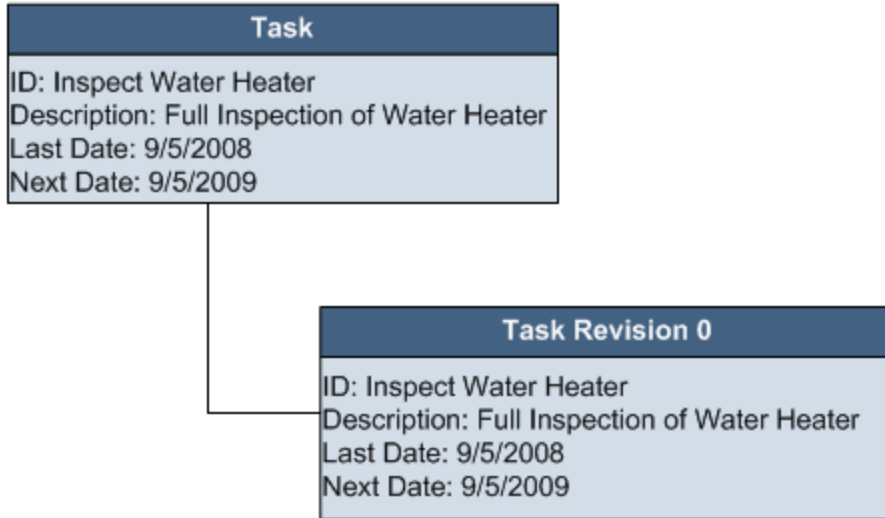
If you later change the Task Description from *Full Inspection of Water Heater* to *Full External Visual Inspection of Water Heater*, a new Task Revision record would be created and linked to the Task record, as shown in the following diagram.

Task

ID: Inspect Water Heater
Description: Full External Visual Inspection of Water Heater
Last Date: 9/5/2008
Next Date: 9/5/2009

Task Revision 0

ID: Inspect Water Heater
Description: Full Inspection of Water Heater
Last Date: 9/5/2008
Next

Task Revision 1

ID: Inspect Water Heater
Description: Full External Visual Inspection of Water Heater
Last Date: 9/5/2008
Next Date: 9/5/2009
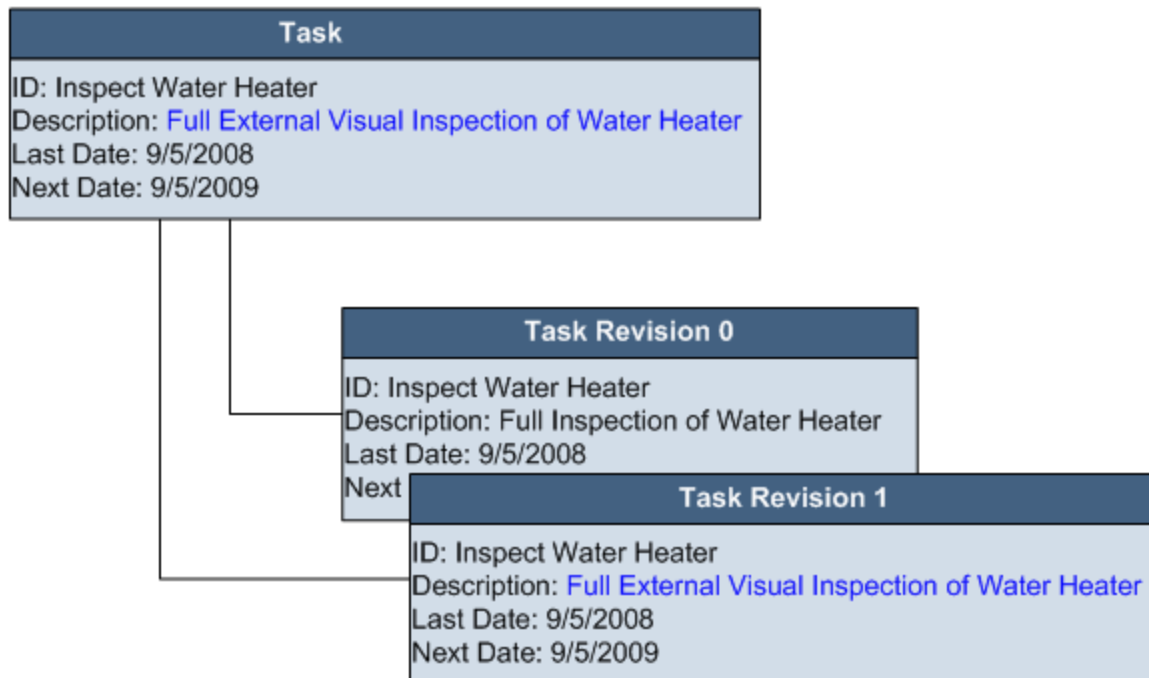
Viewing Revisions of ALL Task Records Linked to a Given Equipment or Functional Location Record

If an Equipment or Functional Location record is linked to multiple Task records, you can view those Task records and their linked Task Revision records.

To view revisions of ALL Task records that are linked to a given Equipment or Functional Location record:

1. In the Meridium APM Framework, on the **Task List** page, select a row containing a Task record that is linked to the desired Equipment or Functional Location record.

2. On the **Asset Options** menu, click the **Show Revisions** link.

   The **Task Revision History** page appears.
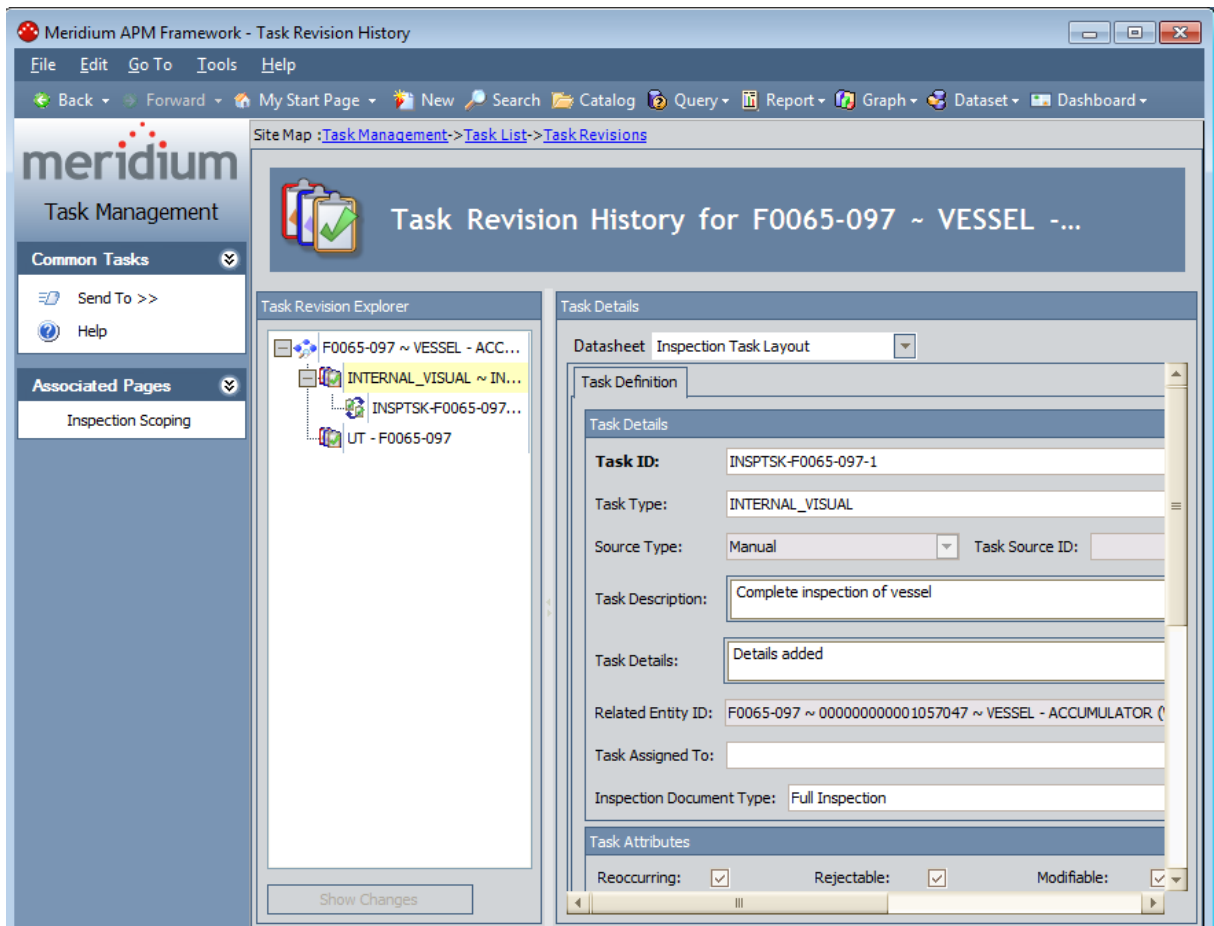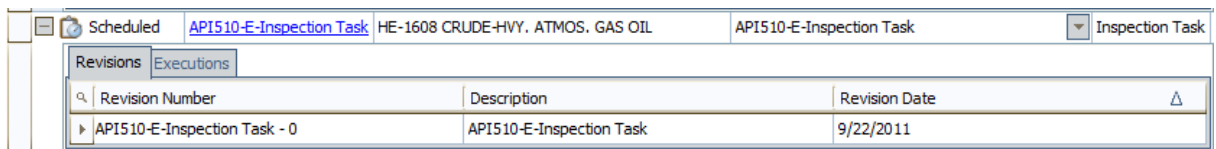
Viewing Revisions of a Single Task Record

To view all revisions of a single Task record:

1. In the Meridium APM Framework, on the **Task List page**, select the row containing the Task record whose revisions you want to view.

2. In the selected row, in the **Current State** column, click the plus sign.

   The **Revisions** tab appears below the selected row. Each row on the **Revisions** tab represents a Task Revision record that is linked to the selected Task record. The records are sorted in descending order according to their Revision Numbers. In the following image, you can see that one Task Revision record is displayed in the grid.

3. To see a list of differences that exist between two Task Revision records, click the hyperlinked Revision Number of any Task Revision record with a Revision Number greater than 0.

   The **Changes window** appears.

Aspects of the Task Revision History Page

The **Task Revision History** page contains the following items:

- **Common Tasks menu:** Provides access to common functionality.
- **Associated Pages menu:** Displays Associated Pages that have been configured for the family of the record that is selected in the **Task Revision Explorer** pane.
- **Task Revision Explorer pane:** Displays an Equipment or Functional Location record, its linked Task records, and their linked Task Revision records.
- **Task Details pane:** Displays the datasheet of the record that is selected in the **Task Revision Explorer** pane.
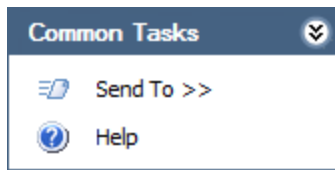
Common Tasks Menu

The **Common Tasks** menu on the **Task Revision History** page contains the following links:

- **Send To:** Displays a submenu with options that let you provide a link to the current page on your desktop (create shortcut), in an email message, or on a Home Page.
- **Print:** Displays the **Preview** window, which displays the datasheet for the record that is selected in the **Task Revision Explorer** pane.
- **Help:** Displays the context-sensitive Help topic for the **Task Revision History** page.

Task Revision Explorer Pane

The **Task Revision Explorer** pane on the **Task Revision History** page contains a tree that displays an Equipment or Functional Location record, its linked Task records, and their linked Task Revision records. The Equipment or Functional Location record that appears in the **Task Revision Explorer** pane is the Equipment or Functional Location record that is linked to the Task record that was selected on the **Task List** page from which you accessed the **Task Revision History** page.

The root level of the tree displays the Equipment or Functional Location record, which is represented by the ✤ icon. The text describing the Equipment or Functional Location record is the Record ID of the Equipment or Functional Location record.

Below the Equipment or Functional Location record are nodes representing the Task records that are linked to the Equipment or Functional Location record. Each Task record is represented by the 🗐 icon. The text describing the Task record is the Record ID of the Task record.

Below the Task records are nodes representing the Task Revision records that are linked to the Task record. Each Task Revision record is represented by the 🖧 icon. The text describing the Task Revision record is the Record ID of the Task Revision record, which by default includes the Task ID and the Revision Number, separated by a dash. For example, if the Task ID is *Insp1-8281*, and the Revision Number is *0*, the tree would display the node shown in the following image.
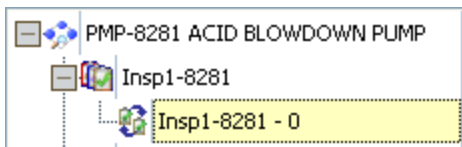
Note that Task Revision records are listed in the tree in descending order according to Revision Number.

Below the tree is the **Show Changes** button. After you select a Task Revision record in the tree that contains a Revision Number greater than 0, you can click the **Show Changes** button to see a [list of the differences that exist between that Task Revision record and the Task Revision record with the next lowest Revision Number](#).

Task Details Pane

The **Task Details** pane on the **Task Revision History** page displays the datasheet of the record that is selected in the **Task Revision Explorer** pane.

The **Task Details** pane is shown in the following image.

Note that you cannot modify any field values in the **Task Details** pane.

Viewing a List of Differences Between Two Task Revision Records

To view a list of differences between two Task Revision records:
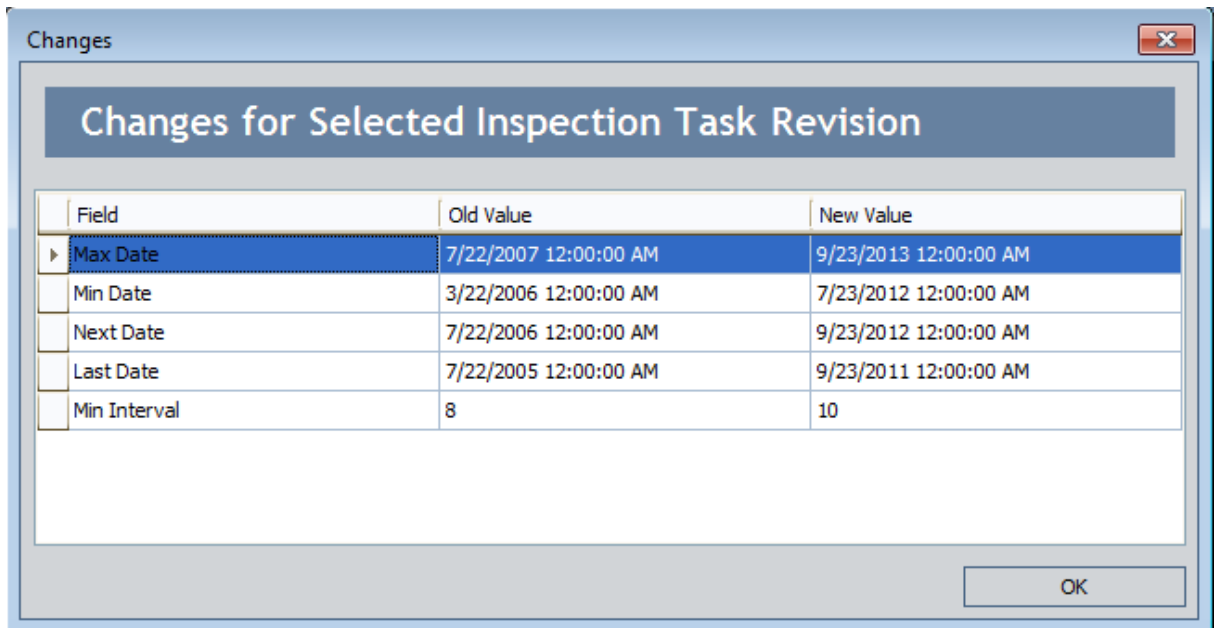
- In the Meridium APM Framework, on the **Task List** page, on a **Revision** tab, click the hyperlinked Revision Number.

  -or-

  In the Meridium APM Framework, on the **Task Revision History** page:

  a. In the tree in the **Task Revision Explorer** pane, select a Task Revision record.

  b. Below the tree, click the **Show Changes** button.

  The **Changes window** appears.

About the Changes Window

The **Changes** window displays a list of differences that exist between a Task Revision record and the Task Revision record with the next lowest Revision Number (that is linked to the same Task record).
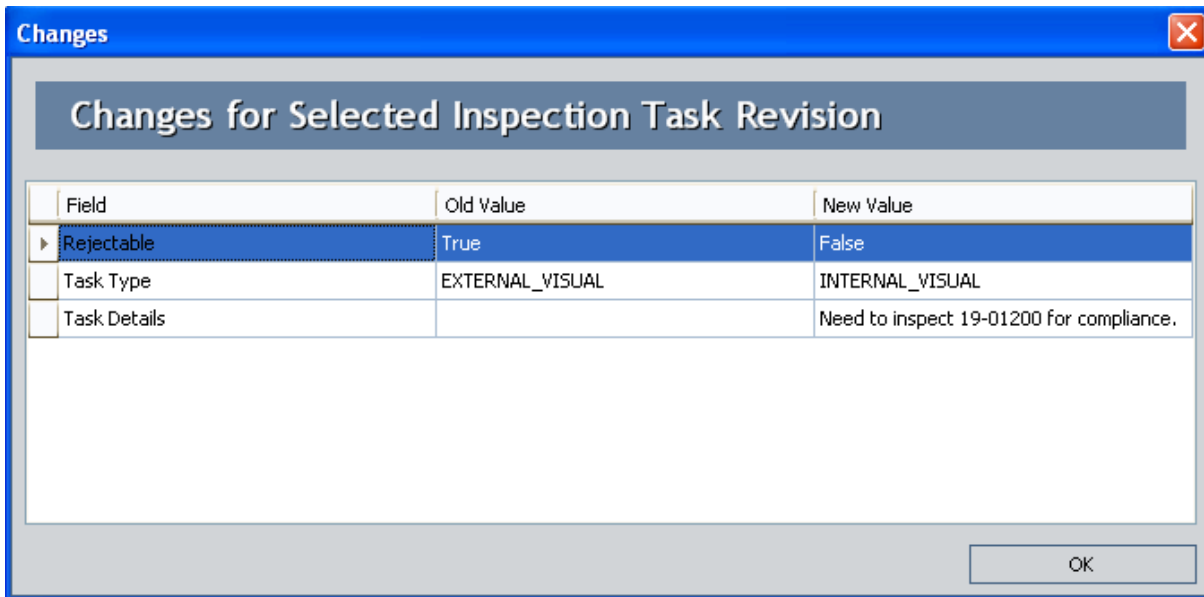
For example, in the previous image, you can see that the following differences exist between this Task Revision record and the last Task Revision record that was created and linked to the same Task record:

- The Rejectable field now contains the value *False*.
- The Task Type field now contains the value *INTERNAL_VISUAL*.
- The Task Details field now contains the value *Need to inspect 19-01200 for compliance*.

You can access the **Changes** window by:

- Clicking the **Show Changes** button on the **Task Revision History** page.
- Clicking a hyperlinked Revision Number on the **Task List** page.

About Task Types Records

Task Types records control which values appear in the **Task Type** list in Task and RBI Recommendation records. Each Task Types record has a *Reference* field, which is used to associate that task type with a specific family (e.g., Inspection Task).

The baseline Meridium APM database contains a predefined set of Task Types records. Default rules are also provided to filter the **Task Type** list in Calibration Task, Inspection Task, and RBI Recommendation records. You can add additional Task Types records and rules if desired.

The following table describes which families use which references to populate the **Task Type** list.

Family	Reference Value Used	Notes
Calibration Task	Calibration	None
Inspection Task	Inspection -or- Inspection_Strategy	In Inspection Task records, the reference value that is used differs depending upon how the RBI administrative settings are configured. • If RBI is not configured to generate RBI Recommendation records using Policies, the **Task Type** list in Inspection Task records comes from Task Types records with the *Inspection* reference. • If RBI IS configured to generate RBI Recommendation records using Policies, the **Task Type** list in Inspection Task records comes from Task Types records with the *Inspection_Strategy* reference. • If RBI is not licensed, the **Task Type** list in Inspection Task records comes from Task Types records with the *Inspection_Strategy* reference.
Task (records in the root Task family)	None	If you create a Task record that belongs to the root Task family, the **Task Type** list will contain the Task Type ID of ALL Task Types records that exist in the database.

Family	Reference Value Used	Notes
RBI Recom-mendation	Inspection -or- Inspection_Strategy	In RBI Recommendation records, the reference value that is used differs depending upon how the RBI administrative settings are configured. • If RBI is not configured to generate RBI Recommendation records using Policies, the **Task Type** list in RBI Recommendation records comes from Task Types records with the *Inspection* reference. • If RBI IS configured to generate RBI Recommendation records using Policies, the **Task Type** list in RBI Recommendation records comes from Task Types records with the *Inspection_Strategy* reference.

Accessing the List of Task Types Records

To access the list of the Task Types records:

1. In the Meridium APM Framework, on the [Task Management Start Page](#), click the **Manage Task Types** link.

   > **Note:** You can view Task Types records only if you have been granted View permissions to the Task Types family.

   The **Find Items** window appears. The **Search In** list is automatically populated with the *Task Types* family. You cannot select another family.
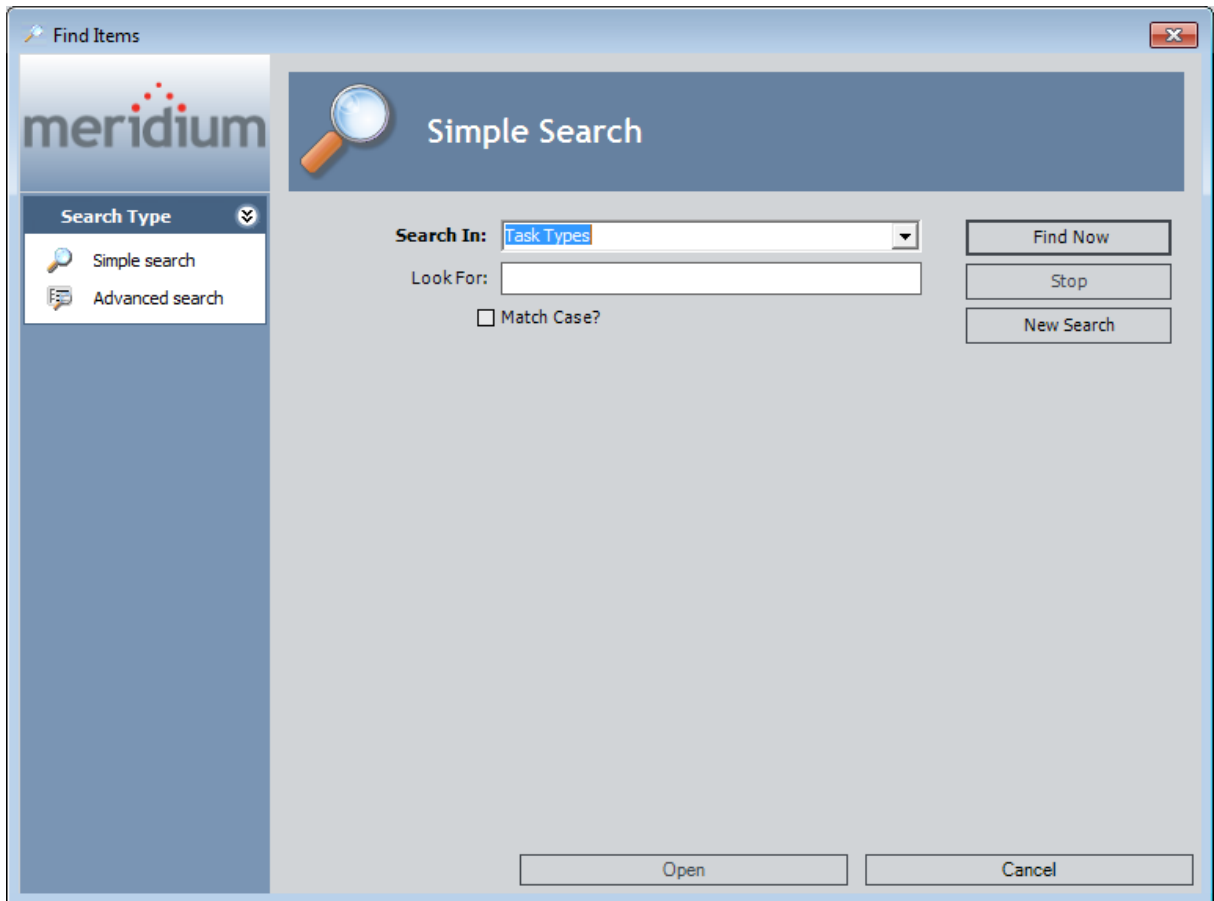
2. Click the **Find Now** button.

3. In the results list, select the record that you want to view, and click the **Open** button.

   -or-

   To view a list of all Task Types records, click **Cancel**.

The **Task Types** page appears, displaying the selected Task Types records.
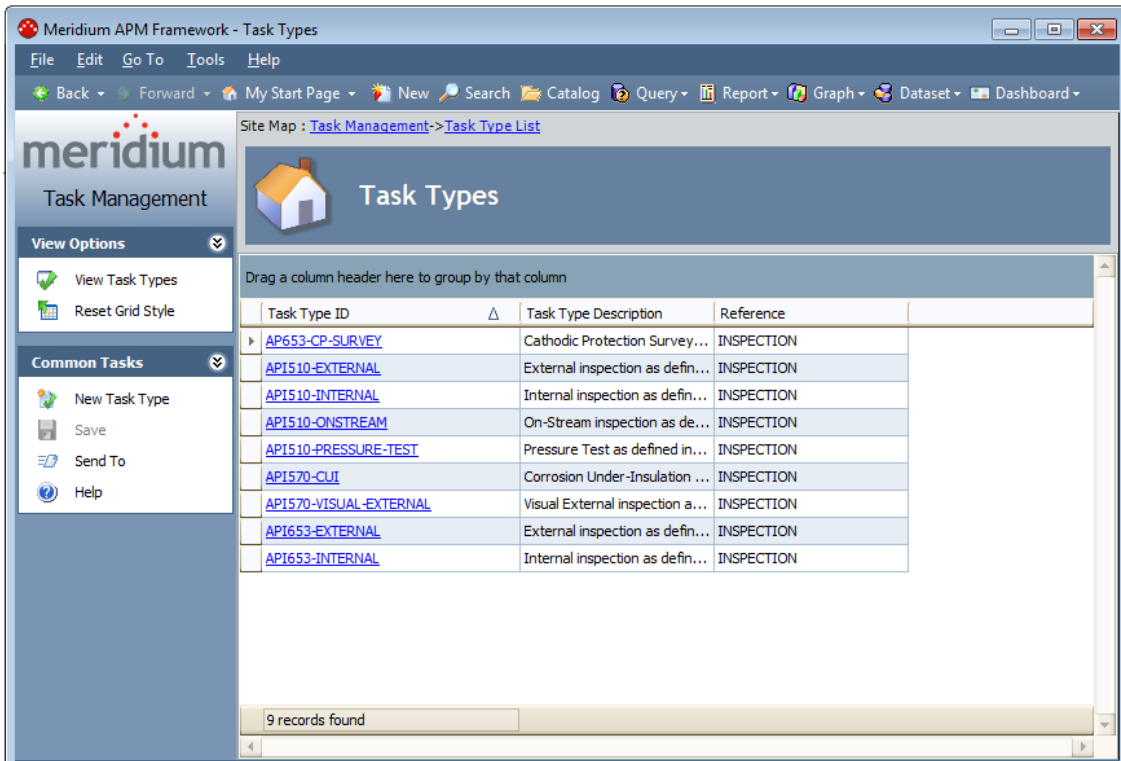
Note: The Task Type IDs displayed in the list are the exact values that appear in the **Task Type** list on the Task datasheet. This list is populated based on the family to which the Task record belongs. For an example, if the Task record belongs to the Inspection Task family, the **Task Type** list will contain only IDs of Task Types records where the Reference field contains the value Inspection.

Aspects of the Task Types Page

The **Task Types** page displays a list of Task Types records according to your selection in the search results when you chose to access the page.

For each Task Types record listed, the following information is displayed:

- **Task Type ID:** The ID assigned to the Task Types record.
- **Task Type Description:** A brief description of the Task Types record.
- **Reference:** The reference assigned to the Task Types record. The reference determines whether the ID of the Task Types record will appear in the **Task Type** list on the Task datasheet for a specific family. For example, if the Reference is Inspection, the ID of the Task Types record will be available in the **Task Type** list when you create an Inspection Task record.

View Options Menu

The **View Options** menu on the **Task Types** page contains the following links:

- **View Task Types:** Displays the **Find Items** window, where you can select the Task Types records that you want to see in the list.

- **Reset Grid Style:** After asking for confirmation, displays the **Find Items** window, where you can select the Task Types records that you want to see in the list. When the records are displayed, the grid is reset to display the default columns.

Common Tasks Menu
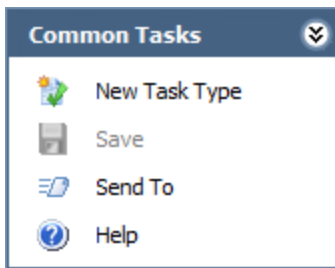
The **Common Tasks** menu on the **Task Types** page contains the following links:
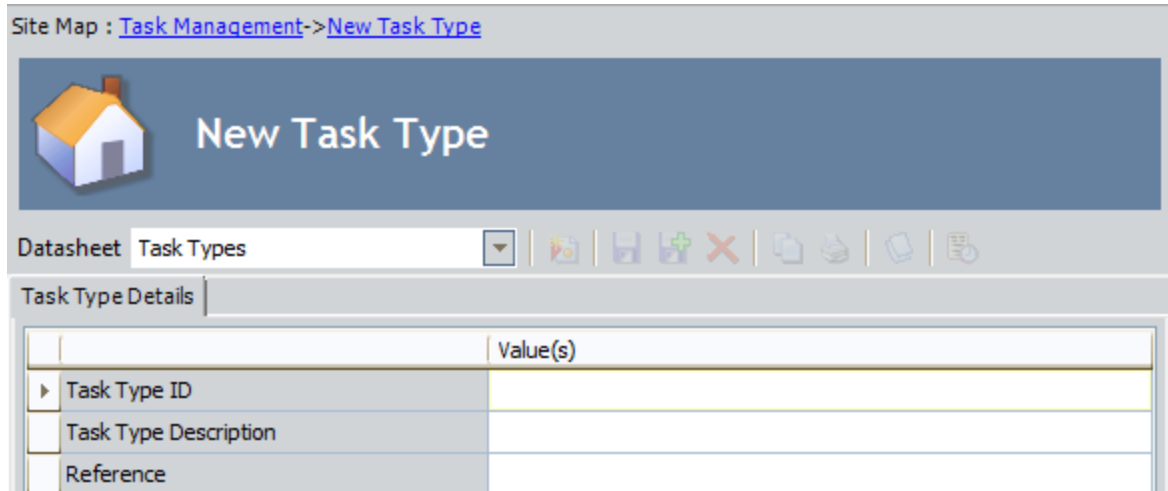
- **New Task Type:** Displays a blank Task Types datasheet on the **Task Types** page.
- **Save:** Saves the Task Types record. This link is disabled until you select a Task Type record to modify and open it in the **Edit Task** form.
- **Send To:** Displays a submenu with options that let you provide a link to the current page on your desktop (create shortcut), in an email message, or on a Home Page.
- **Help:** Displays the context-sensitive Help topic for the **Task Types** page.

Creating a New Task Types Record

To create a new Task Types record:

1. In the Meridium APM Framework, on the **Task Types** page, click the **New Task Type** link.

   A blank Task Types datasheet appears.

2. In the **Task TypeID** text box, type the ID of the new Task Types record.

3. In the **Task Type Description** box, type a description for the new Task Types record.

4. In the **Reference** list, select a reference for the new Task Types record. Your selection determines the values that appear in the **Task Type** list on the Task datasheet of a specific family. For example, if you select **Inspection** as the Reference, the ID of this new Task Types record will be available in the **Task Type** list when you create an Inspection Task record.

5. On the **Common Tasks** menu, click the **Save** link.

   The record is saved to the database.

Modifying an Existing Task Types Record

To modify an existing Task Types record:

1. In the Meridium APM Framework, access the list of Task Types records.

2. On the **Task Types** page, click the hyperlinked Task Type ID for the Task Types record you want to modify.

   The Task Types datasheet appears.
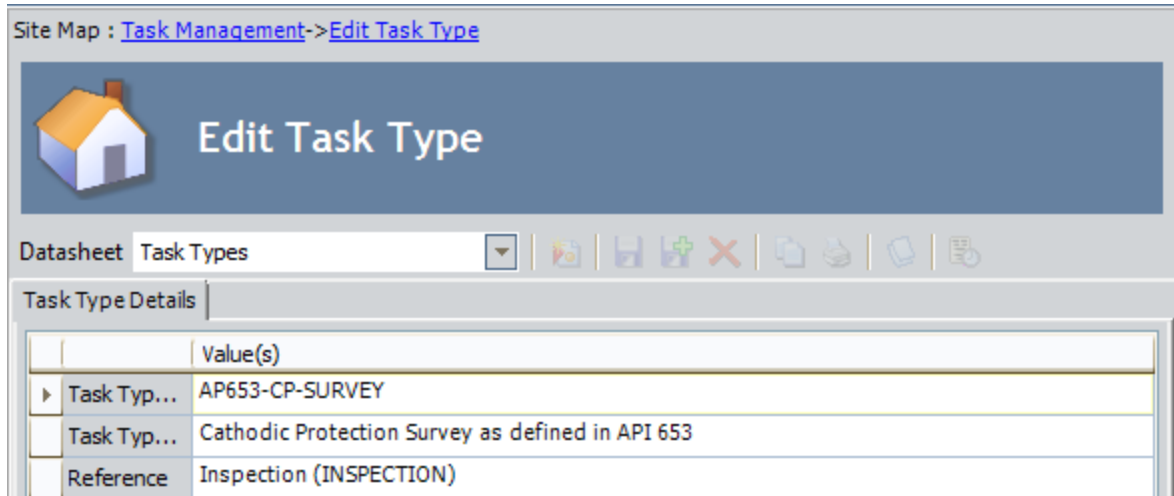
3. In the **Task TypeID** text box, modify the ID of the Task Types record as desired.

4. In the **Task Type Description** box, modify the description for the Task Types record as desired.

5. In the **Reference** list, modify the reference for the Task Types record as desired.

6. On the **Common Tasks** menu, click the **Save** link.

   The record is saved to the database.

About the Timeline

The Timeline is a visual representation of Scheduled Task records. The Timeline displays the following field values of each Task record that appears on the Timeline:

- Task ID
- Last Date
- Min. Date
- Max. Date
- Next Date
- Desired Interval

You can filter the Timeline to display only Task records whose minimum and maximum dates fall within a certain date range. You can also display global event records on the Timeline for the purpose of comparing Task record Next Dates against a global event record date that you define when setting up a global event query. You can use the visual comparison of dates to determine whether or not to change the Next Date of a Task record so that it matches the date of the global event record.

A global event record is simply a record that belongs to any family or subfamily. The record should represent some major occurrence in your business operation, such as a shutdown or turnover, but it does not have to belong to any certain family, such as the Event family. You can define any family of your choice in which to store global event records.

Accessing the Timeline

To access the Timeline from the Task Management Start Page:

1. In the Meridium APM Framework, on the <u>Task Management Start Page</u>, click the **Manage Timeline** link.

   The <u>Task Query Builder</u> appears.
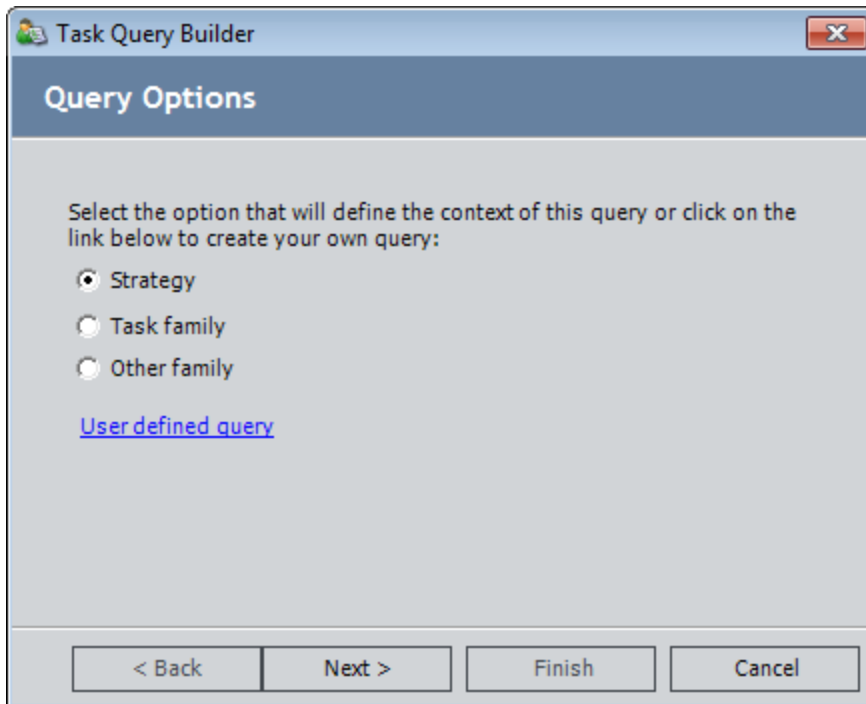
2. Proceed through the **Task Query Builder** to select the Task records that you want to view on the **Timeline** page.

   The <u>Timeline page</u> appears.

To access the Timeline from the Task List page:

- On the <u>Task List page</u>, on the **Optimize Tasks** menu, click the **Optimize Time Line** link.

   The <u>Timeline page</u> appears.

Aspects of the Timeline Page

When you access the **Timeline** page, Task records are represented on the Timeline by blue diamonds, as shown in the following image. A red horizontal line runs through each blue diamond to indicate the date range of each Task record. The Minimum Date and Maximum Date for each Task record are represented by short, vertical lines at either end of each red line. The Minimum Date and Maximum Date cannot be changed via the Timeline. The position of each blue diamond represents the Next Date of that Task record, relative to the date range displayed on the Timeline.

The x-axis of the Timeline displays the start date and the end date of the Timeline. The y-axis of the Timeline displays the Task IDs of the Scheduled Task records that are displayed.

You can change the appearance of the Timeline by filtering the Task records that are displayed based on a particular date range.

If you previously snapped a Task record to a global event record and the Timeline is displaying that Task record, the linked global event record will not be displayed. You can display the global event record again via the **View Options** menu.

If you choose to display a global event record on the Timeline via the **View Options** menu, it will be represented by a vertical line. The line will be yellow if the global event has been activated, meaning you can snap multiple Task records to it. It will be green if it has not been activated.

If you snap a Task record to a global event record, the diamond representing the Task record will change from blue to green and will lie directly on the line representing the global event record, as shown in the following image.

View Options Menu

The **View Options** menu on the **Timeline** page contains the following links:

- **Overlay Global Events:** Displays the **Global Events** dialog box, where you can choose which global events to overlay on the Timeline.
- **Change Time Base:** Displays the **Time Base Properties** dialog box, where you can change the Start Date, End Date, Scale Number, and Scale Period.
- **Filter Tasks:** Displays the **Filter Tasks** dialog box.

  > **Note:** The **Filter Tasks** dialog box does not function as intended and, therefore, is not documented. If you use the **Filter Tasks** dialog box and see unexpected results, you can click the **Reset Timeline** link on the **View Options** menu to return the Timeline to its original display.

- **Snap Tasks:** Displays the **Snap Tasks** dialog box, where you can select multiple Task records whose dates you want to coincide with the date of the global Event record.
- **Unsnap Tasks:** Displays the **Unsnap Tasks** dialog box.

  > **Note:** The **Unsnap Tasks** dialog box does not function as intended and, therefore, is not documented. If you use the **Unsnap Tasks** dialog box to remove a link between a Task record and a global event record and then save your changes, the changes will appear saved from within the Timeline. If, however, you access the unsnapped Task record in the Record Manager, you will still see the link between the Task record and the global event record. To remove a link between a Task record and a global event record, you should use the Record Manager.

- **Reset Timeline:** Allows you to return all settings to their original values.

Common Tasks Menu
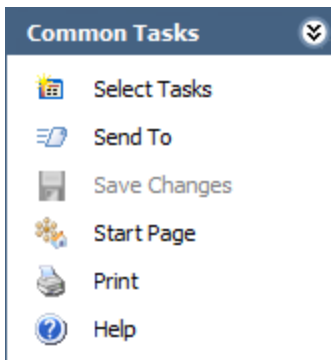
The **Common Tasks** menu on the **Timeline** page contains the following links:

- **Select Tasks:** Displays the **Task Query Builder**, where you can select which Task records to view.
- **Send To:** Displays a submenu with options that let you provide a link to the current page on your desktop (create shortcut), in an email message, or on a Home Page.
- **Save Changes:** Saves any changes you made Task record dates while working with the Timeline.
- **Start Page:** Displays the **Task Management Start Page**.
- **Print:** Prints the Timeline.
- **Help:** Displays the context-sensitive Help topic for the **Timeline** page.

About the Task Query Builder

You can use the **Task Query Builder** to select the Task records that you want to display on the Timeline. Using the **Task Query Builder**, you can:

- [Display all Task records that were generated from a strategy rule](#).
- [Display all Task records that belong to a Task family](#).
- [Display all Task records that are linked to a specific record](#).
- [Display a list of Task records based on a stored query](#).
- [Display a list of Task records based on a new query](#).

Displaying All Task Records That Were Generated From a Strategy Rule

To display all Task records that were generated from a strategy rule:

1. In the Meridium APM Framework, on the Task Management Start Page, click the **Manage Timeline** link.

   The **Task Query Builder** appears, displaying the **Query Options** screen.

   ![Task Query Builder Query Options screen]
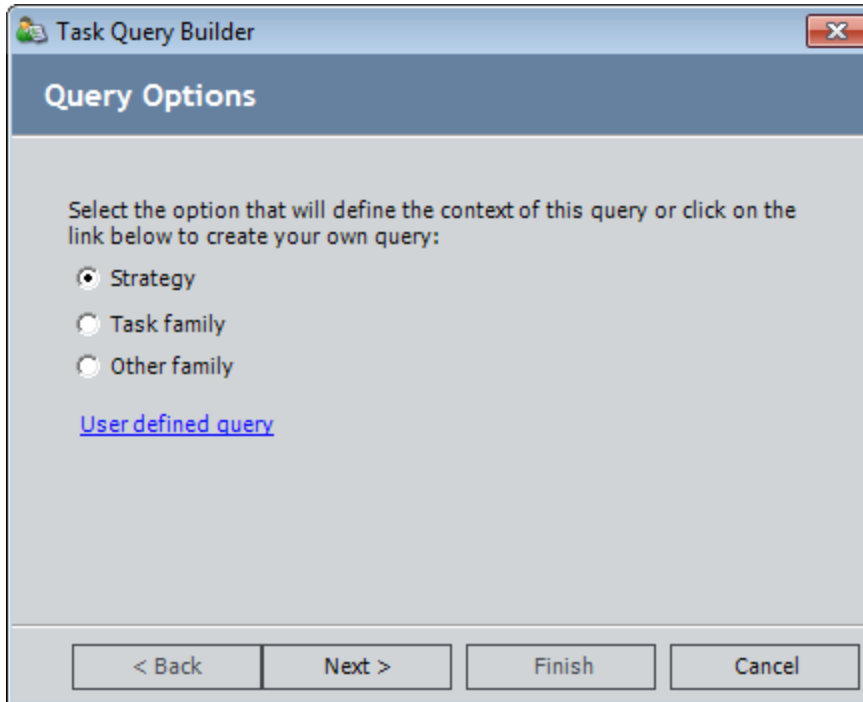
2. Accept the default option, **Strategy**, and click the **Next** button.

   The **Build Task Query** screen appears.
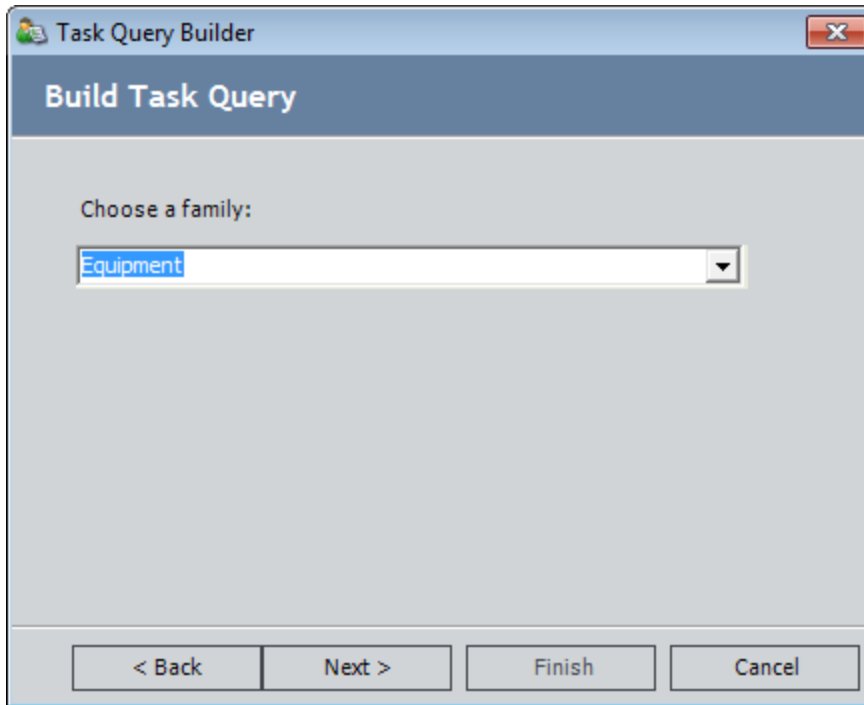
3. In the **Choose a family** list, select the family with which the desired strategy rule is associated. For example, if you are trying to generate a list of Task records that were generated as the result of the execution of Strategy A, and Strategy A is associated with the Pump family, you would select the Pump family here. You will see only the families for which you have been granted at least View permissions in the Configuration Manager.

4. Click the **Next** button.

   The **Strategy** screen appears.
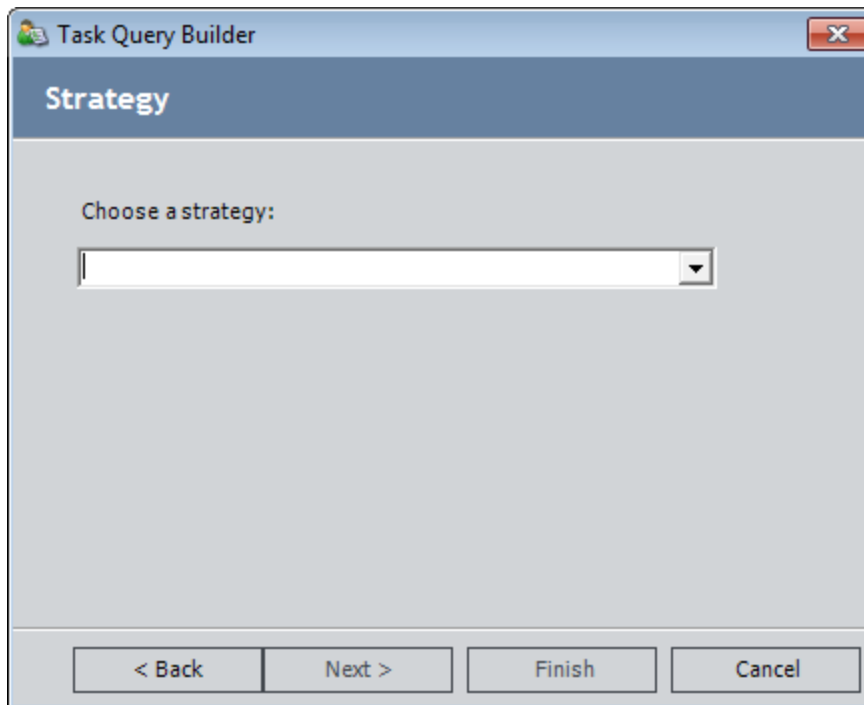
5. In the **Choose a strategy** list, select the Strategy record that contains the desired strategy rule.

6. Click the **Finish** button.

    The list of Task records that were generated as a result of the execution of the strategy rule appears on the **Timeline** page.

Displaying All Task Records That Belong to a Task Family

To display all Task records that belong to a Task family:

1. In the Meridium APM Framework, on the **Task Management Start Page**, click the **Manage Timeline** link.

   The **Task Query Builder** appears, displaying the **Query Options** screen.
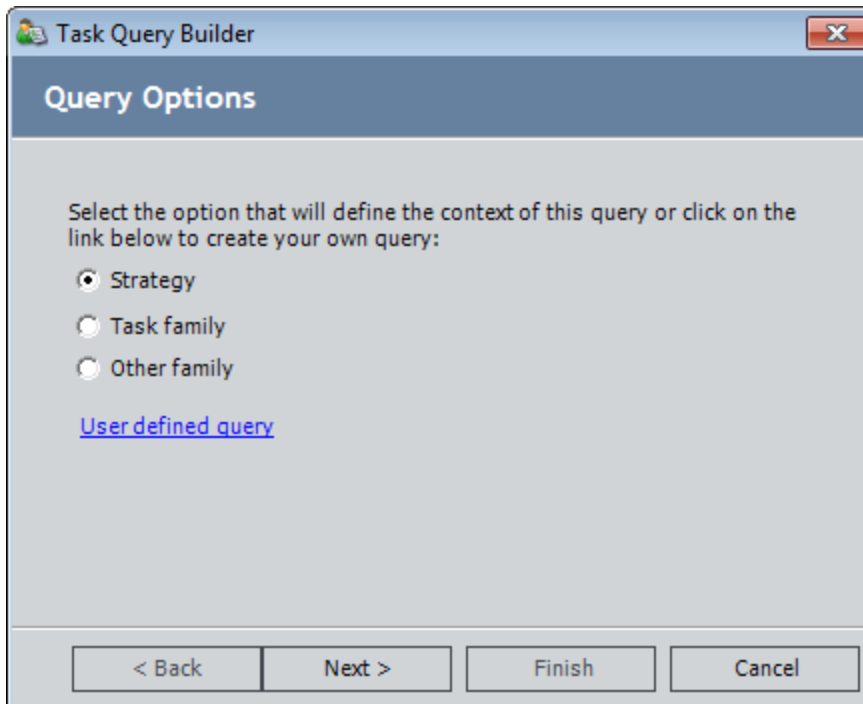
2. Select the **Task family** option, and click the **Next** button.

   The **Task Family** screen appears.
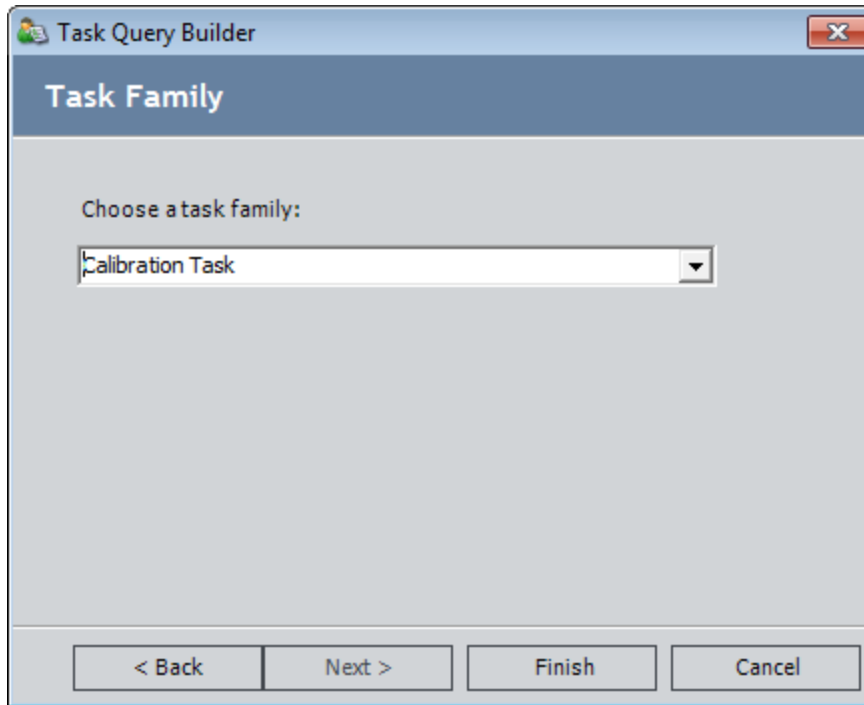
3. In the **Choose a task family** list, select the Task family for which you want to view Task records. You will see only the families for which you have been granted at least View permissions in the Configuration Manager.

4. Click the **Finish** button.

   The list of Task records that belong to the selected Task family appears on the **Timeline** page. The header text displays the selected Task family.

   Note: If you selected to view Task records that belong to the Task family, you will see Task records that belong to the Task family and all of its subfamilies (e.g., the Inspection Task family). You can view an individual Task record's data-sheet, however, only for a Task record that belong to a family to which you have been granted security permissions.

Displaying All Task Records That Are Linked To a Selected Record

To display all Task records that are linked to a selected record:

1. In the Meridium APM Framework, on the [Task Management Start Page](#), click the **Manage Timeline** link.

    The **Task Query Builder** appears, displaying the **Query Options** screen.
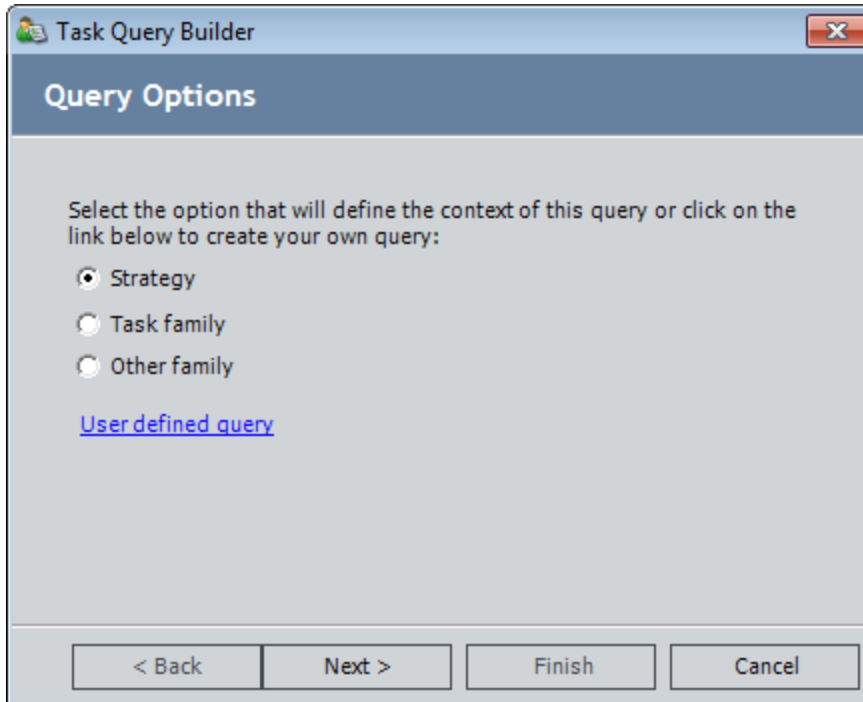
2. Select the **Other family** option, and click the **Next** button.

    The **Build Task Query** screen appears.
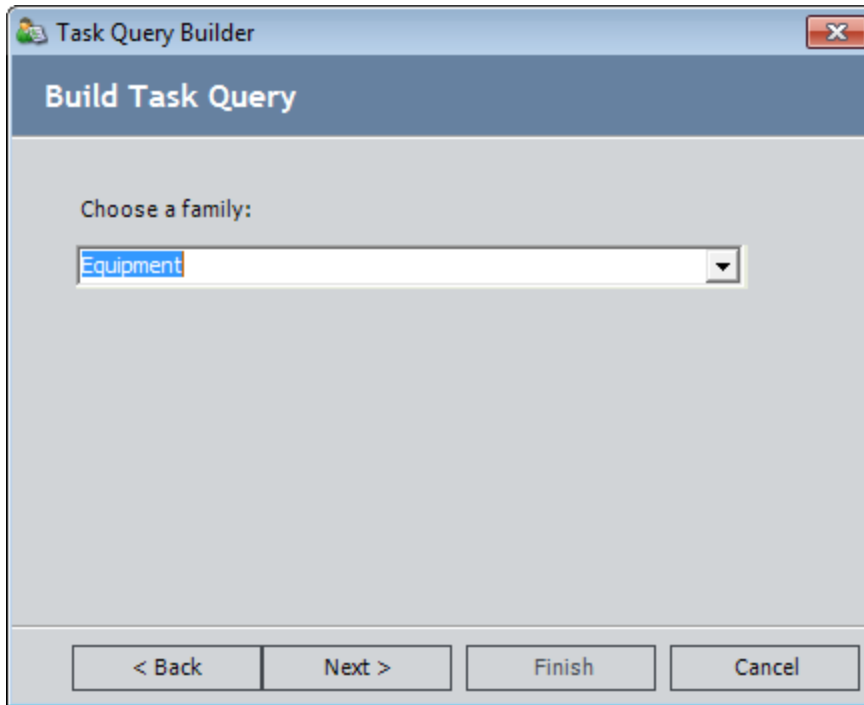
3. In the **Choose a family** list, select the family of the record to which the Task records you want to view are linked. You will see only the families for which you have been granted at least View permissions in the Configuration Manager.

4. Click the **Next** button.

The **Entity** screen appears.
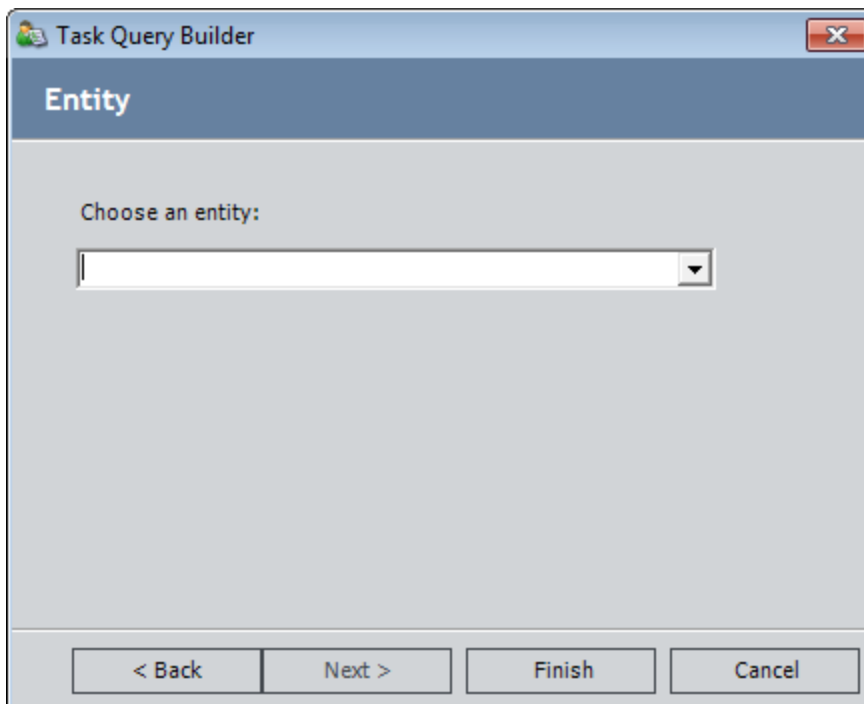
5. In the **Choose an entity** list, select a record whose linked Task records you want to view.

6. Click the **Finish** button.

   The list of Task records that are linked to the selected record appears on the **Task List** page. The header text displays the selected record.

Displaying Task Records Based on a Stored Query

Before you can specify for the Timeline to display Tasks records based on a stored query, you must create the query and save it to the Catalog.

To display Task records based on a stored query:

1. In the Meridium APM Framework, on the [Task Management Start Page](#), click the **Manage Timeline** link.

   The **Task Query Builder** appears.
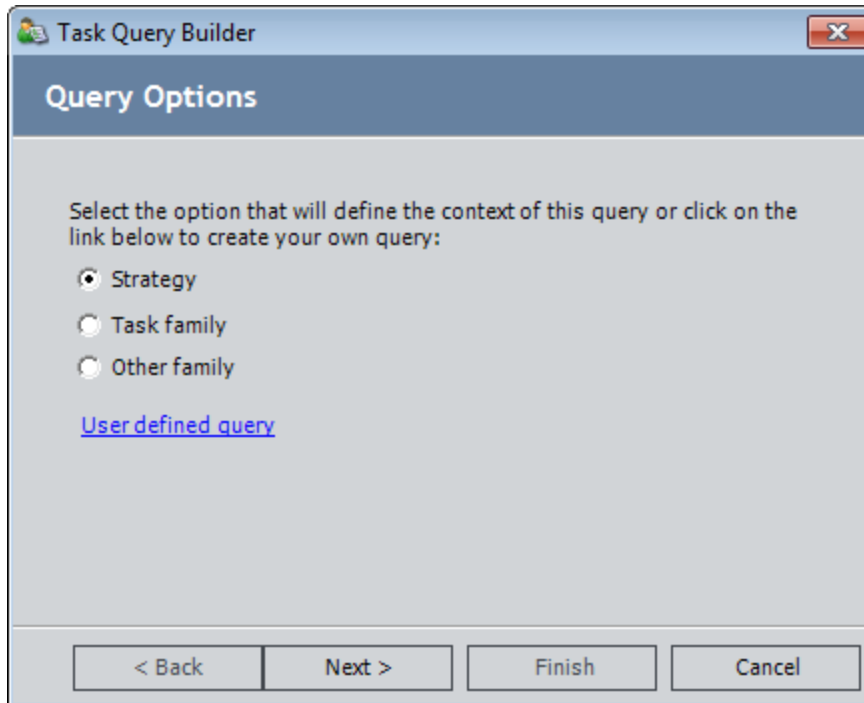
2. Click the **User defined query** link.

   The **User Defined Query Builder** appears, displaying the **Task Query** screen.
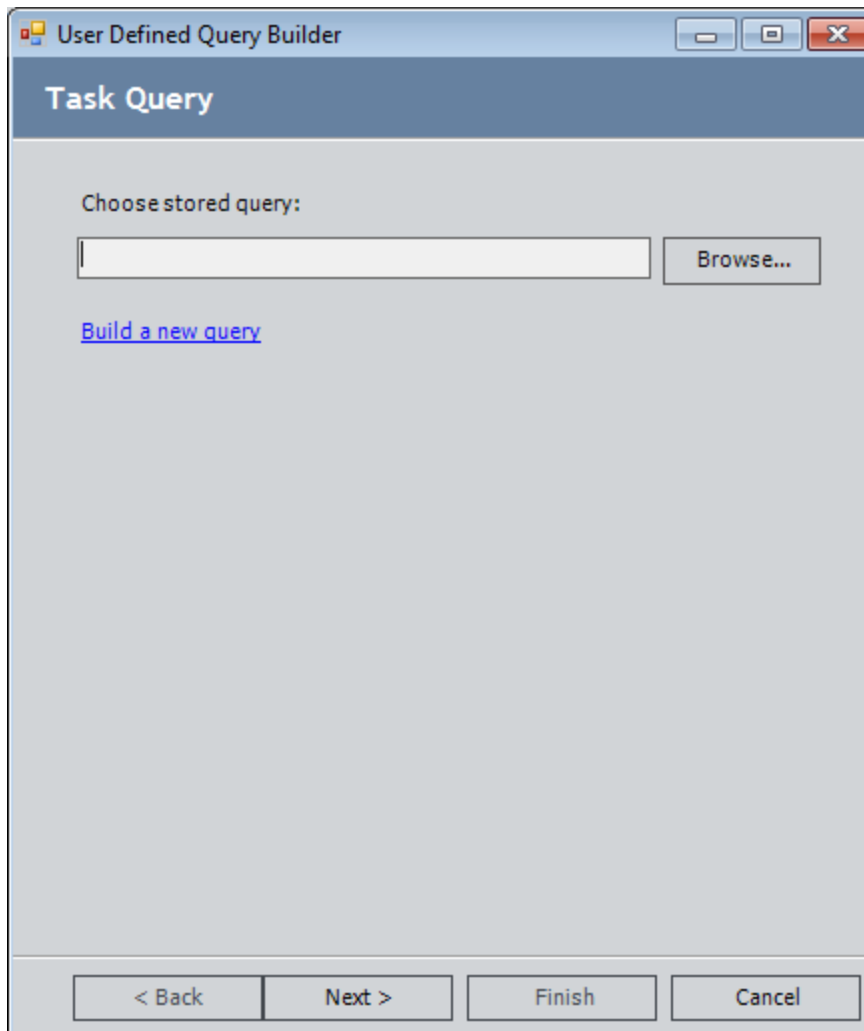
3. Click the **Browse** button to search for a stored query.

   The **Open Query** screen appears.
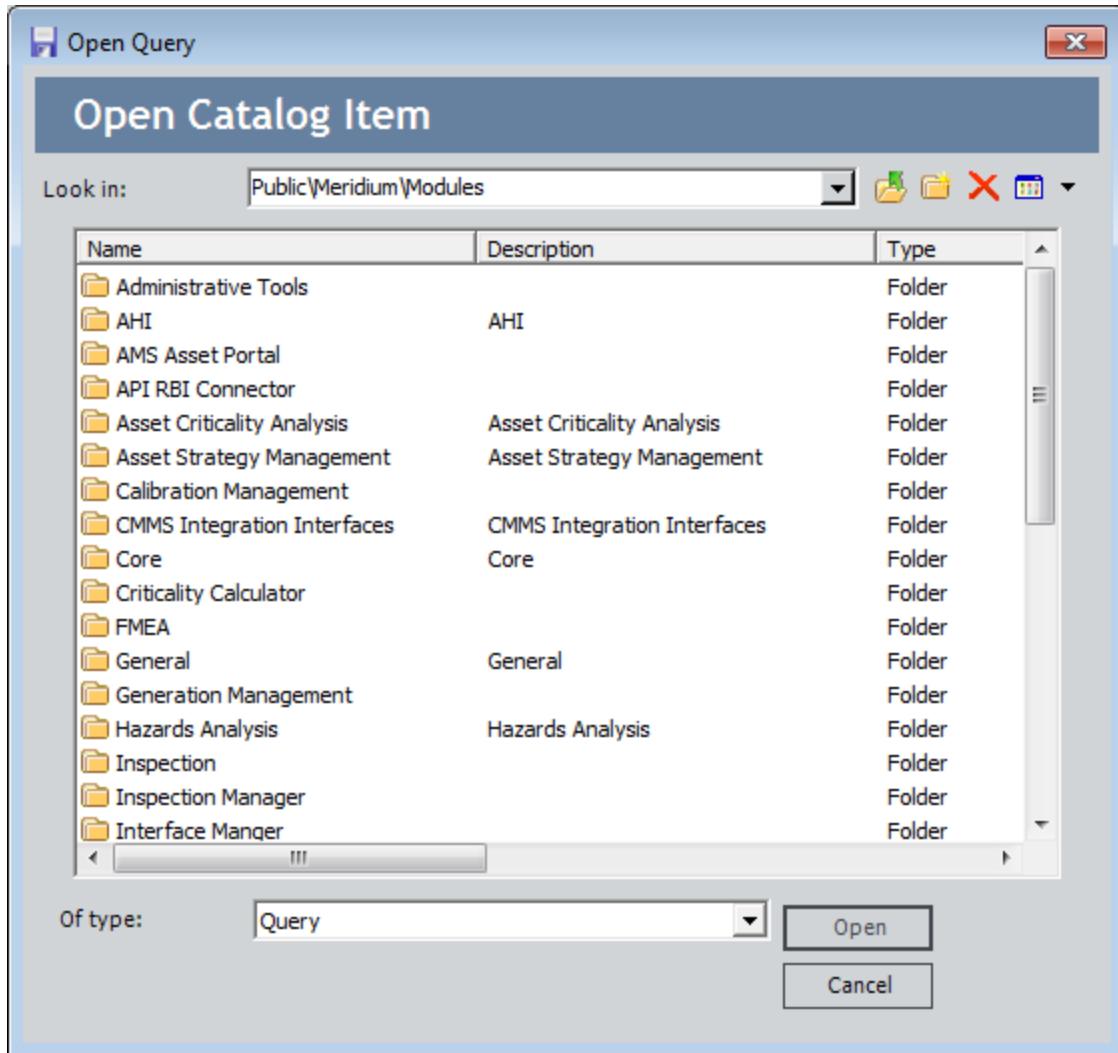
4. Navigate to the appropriate folder and open the desired query.

The **User Defined Query Builder** appears again. The **Choose a stored query** text box is populated with the name of the query you selected.
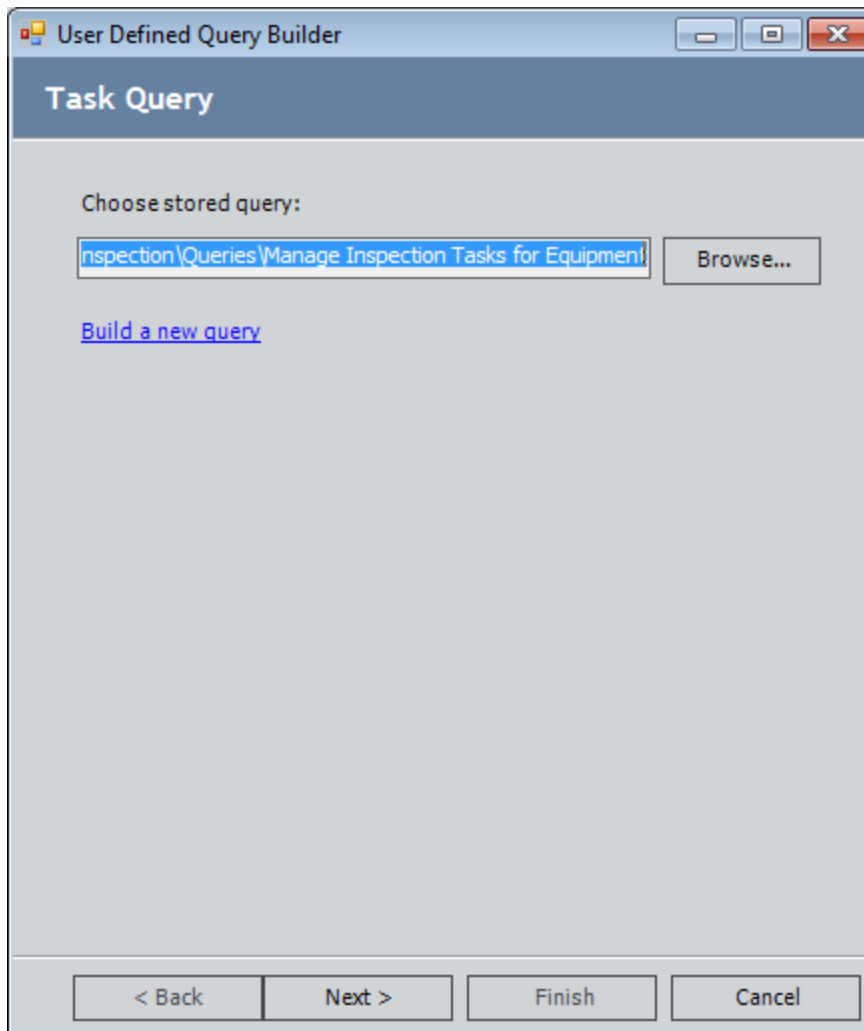
5. Click the **Next** button.

   The **User Defined Query Builder** displays the **Task Query** screen.

6. In each list, select the query field that stores the data that is identified by the list label.

7. Click the **Finish** button.

The Task records from the selected query are displayed on the **Timeline** page.

Displaying Task Records Based on a New Query

To display Task records based on a new query:

1. In the Meridium APM Framework, on the [Task Management Start Page](#), click the **Manage Timeline** link.

    The **Task Query Builder** appears.
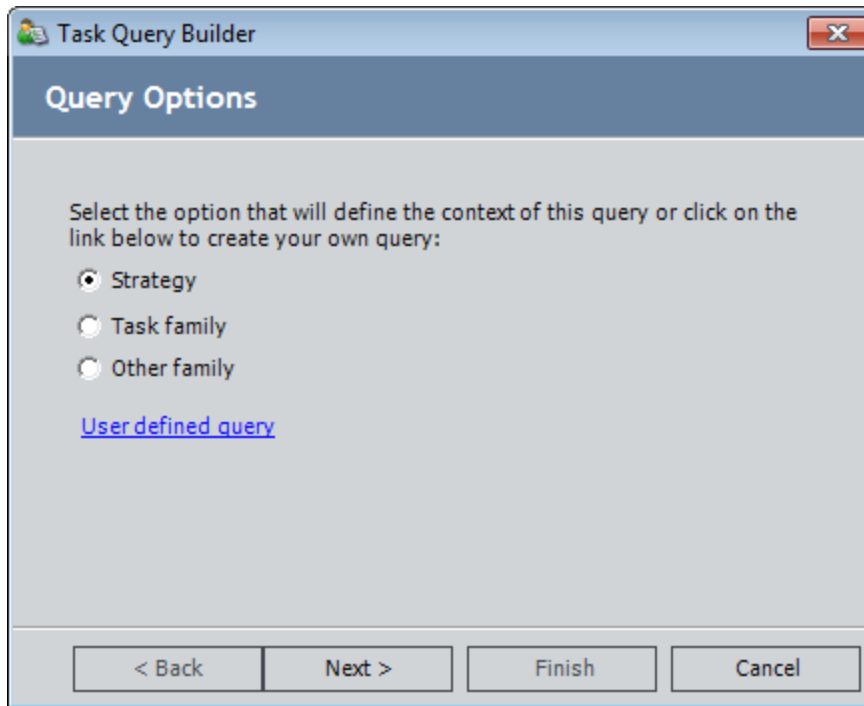
2. Click the **User defined query** link.

    The **User Defined QueryBuilder** appears, displaying the **Task Query** screen.
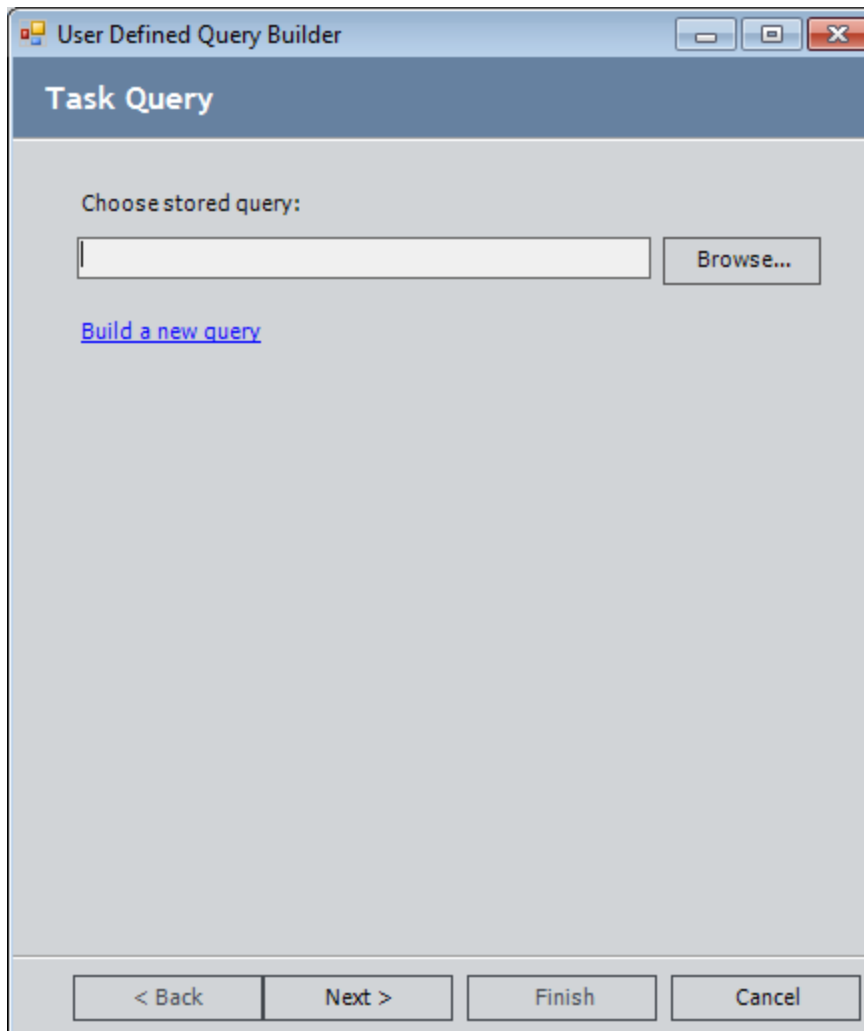
3. Click the **Build a new query** link.

   The **Query Builder** appears.
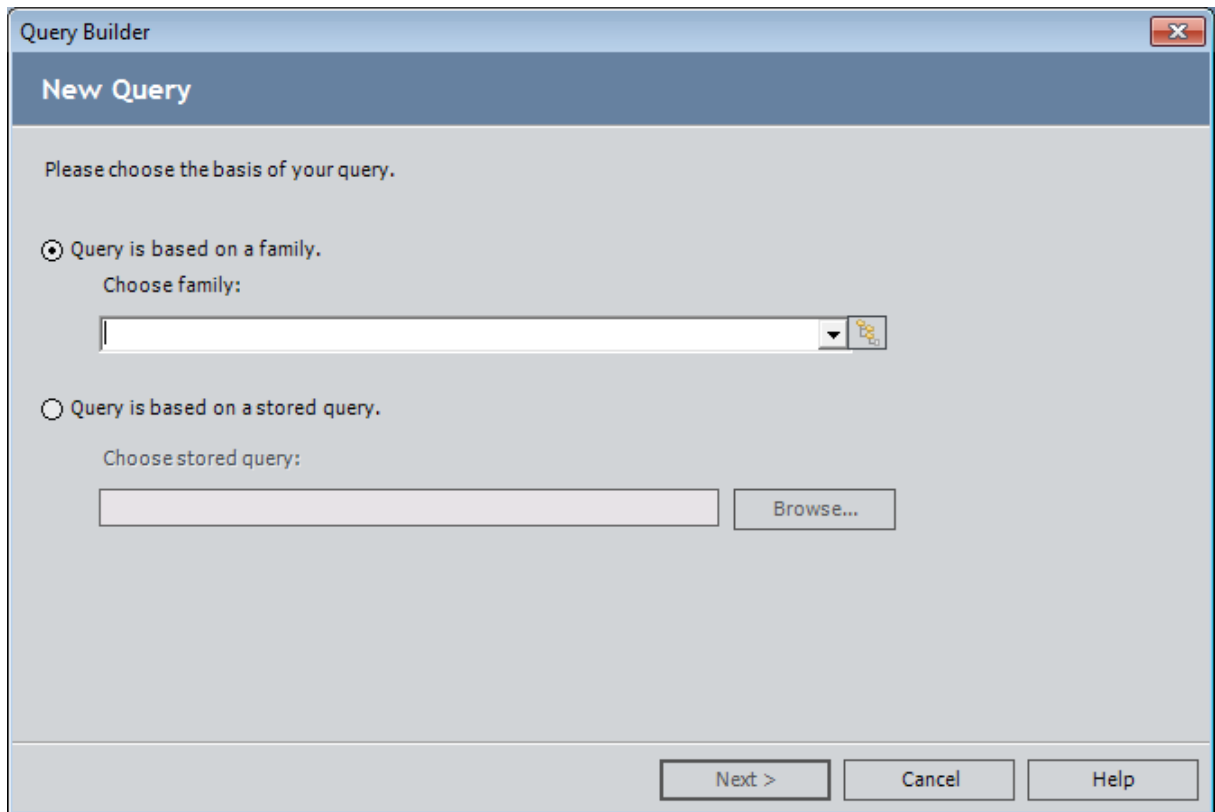
4. Proceed through the **Query Builder**, and click the **Finish** button.

   The Task records from the created query appear on the **Timeline** page.

Displaying Values of a Task Record

To display values of a Task record on the Timeline:

- In the Meridium APM Framework, on the **Timeline** page, position your mouse pointer over a diamond symbol, which represents the Next Date value of each Task record.

  An information box appears, displaying the Task ID, Last Date, Minimum Date, Maximum Date, Next Date, and Interval.
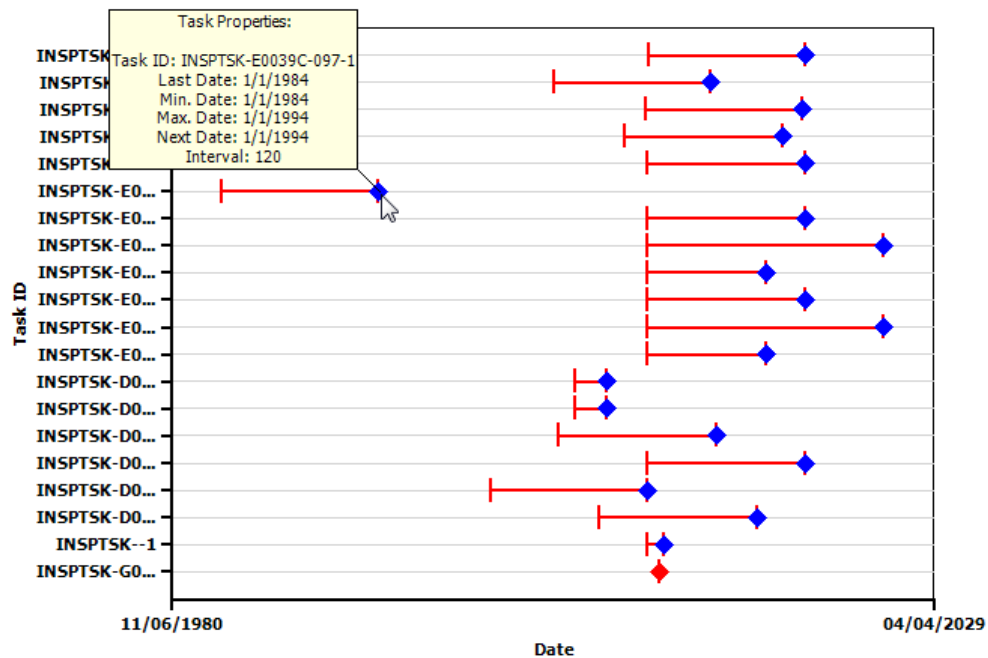
Note that the Minimum Date and the Maximum Date are task constraints. In other words, the Task record's Next Date cannot occur before the Minimum Date or after the Maximum Date on the Timeline.

> **Note:** Task records cannot be modified via the Timeline.

About Changing the Date Range

The Timeline displays Task records that fall within a particular date range, as determined by the Minimum and Maximum dates in the Task record. The range is set to start with the earliest Minimum Date and end with the greatest Maximum Date of all of the Task records you selected for display. The minimum range is one month.
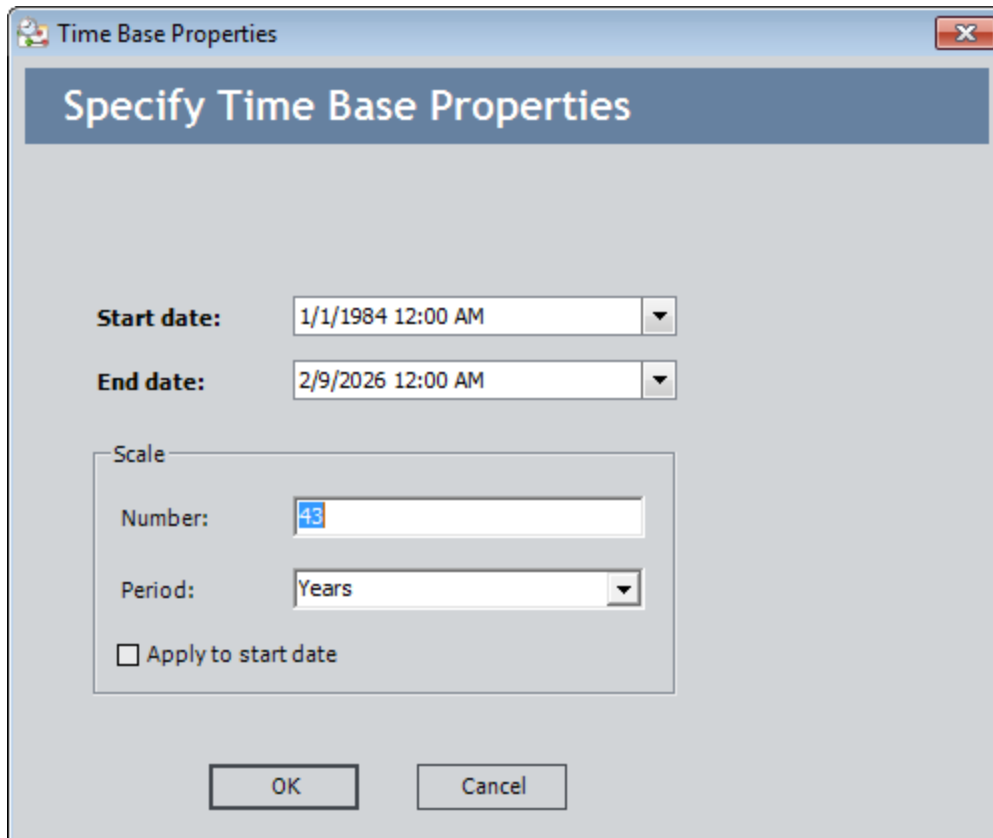
You can change the date range to filter the Task records that are displayed. For example, you could choose to display only Task records where the Minimum Date is on or after January 1, 2005, and the Maximum Date is on or before December 31, 2005.

Changing the Date Range by Specifying a Start Date and End Date

To change the date range by specifying a Start Date and End Date:

1. In the Meridium APM Framework, on the **Timeline** page, on the **View Options** menu, click the **Change Time Base** link.

   The **Time Base Properties** dialog box appears.

2. In the **Start date** box, type or select the start date. The date you select applies to the Minimum Dates of the Task records. So, if you want to display only Task records where the Minimum Date is on or after January 1, 2005, select January 1, 2005, from the Calendar.

3. In the **End date** box, type or select the end date. The date you select applies to the Maximum Dates of the Task records. So, if you want to display only Task records where the Maximum Date is on or before December 31, 2005, select December 31, 2005, from the Calendar.

   The dates you select are reflected in the **Scale** section. For example, if you select January 1, 2005, and December 31, 2005, as the Start date and End date, the value

in the **Number** field will be **12** to indicate the 12 months between January and December.

4. Click **OK** to apply the changes and close the dialog box.
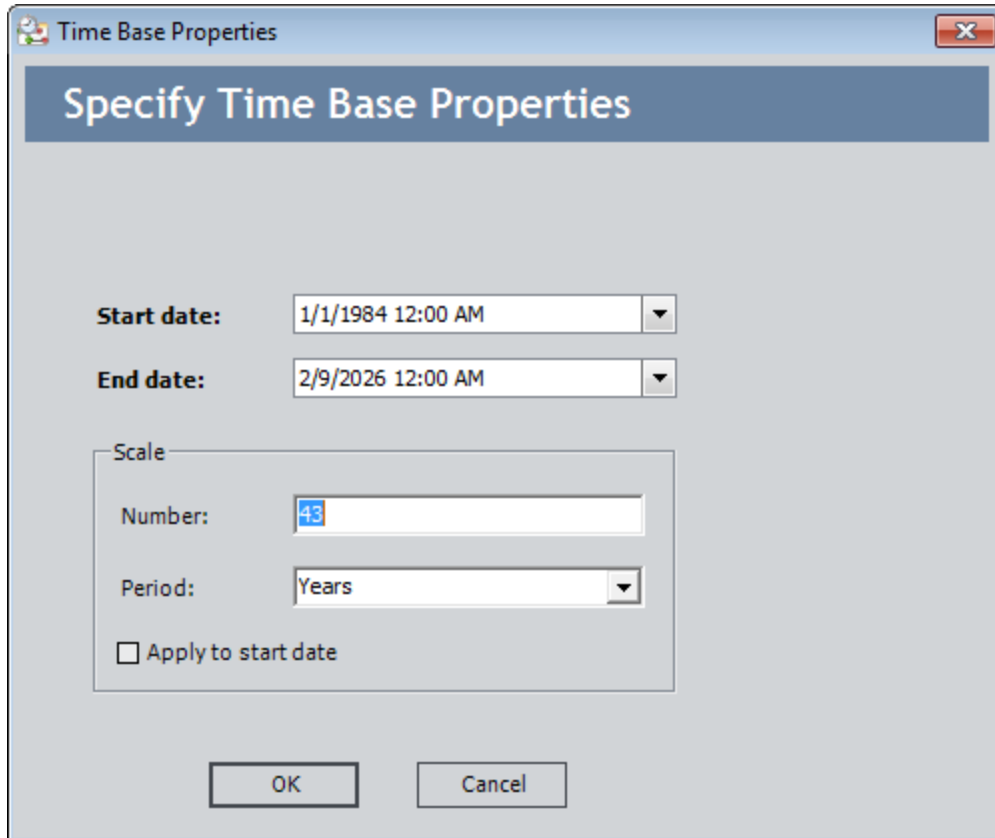
   The parameters you selected are applied to the Timeline.

Changing the Date Range by Defining the Scale

To change the Timebase by Scale Number and Scale Period:

1. In the Meridium APM Framework, on the **Timeline** page, on the **View Options** menu, click the **Change Time Base** link.

   The **Time Base Properties** dialog box appears.

2. In the **Number** text box, type a number to select a period of time over which to display the Task records on the Timeline.

   Note: When you change the number, the **End date** field changes as a result.

3. In the **Period** list, select a different period: either **Months** or **Years**.

   Note: When you change the period, the **End date** field changes as a result.

   Note that if you change the scale number or scale period, the **End Date** will be updated automatically to reflect these changes. If you want to update the **Start Date** instead of the **End Date** based on the scale number or scale period, select the **Apply to start date** check box.

4. Click **OK** to apply the changes and close the dialog box.

   The parameters you selected are applied to the Timeline.

About Displaying Global Event Records on the Timeline

A *global event* record is a record that belongs to any family or subfamily. The family to which the global event record belongs can be a baseline family, such as the Event family, or a family that you define at any root or sublevel. Global event records should represent major occurrences in your business operation, such as a shutdown.

You can choose one or more global event records to display on the Timeline. Global event records are represented as green vertical lines on the Timeline.

To display global event records, you must generate a list of global event records using a global event query, which you define via the **Global Events** dialog box. You can use a stored query to generate a list of global event records, or you can build a new query to generate global event records.

Any query configured to return global event records must contain at least one character field and one date field. You must specify from the query the date field that you want to use as the global event date, which will be used when comparing Task records against the global event record.

Choosing a Stored Query to Generate a List of Global Event Records

To choose a stored query to generate a list of global event records:

1. In the Meridium APM Framework, on the **Timeline** page, on the **Common Tasks** menu, click the **Overlay Global Events** link.

    The most recently selected global event query runs, and the **Global Events** dialog box appears, displaying the records returned by the query.

> **Note:** If no records are returned by the current global event query or a query has not yet been selected, you will receive a warning message. Click **OK** to display the **Global Events** dialog box.

2. Click the **Change global event query** link.

    The **Global Event Query Selection Builder** appears.

3. Click the **Browse** button.

    The **Open a Saved Query** dialog box appears.
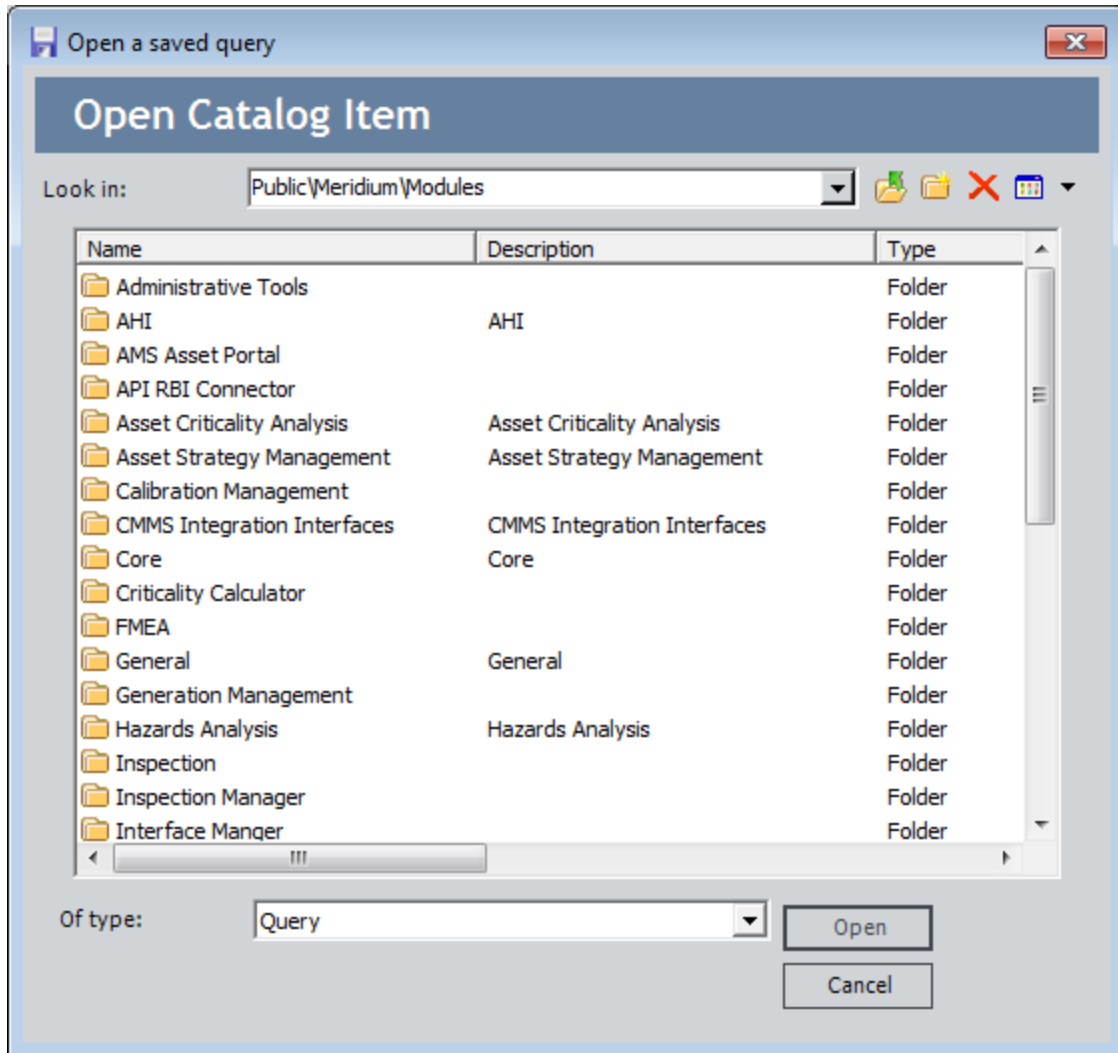
4. Navigate to the folder containing the desired query, select the query, and click the **Open** button. You can select any query that will return the desired records.

> **Note:** The query that you select must contain at least one character column and one date column. If you do not select a valid query, you will receive a message prompting you to select another query.

The **Global Event Query Selection Builder** screen reappears, displaying your selection.
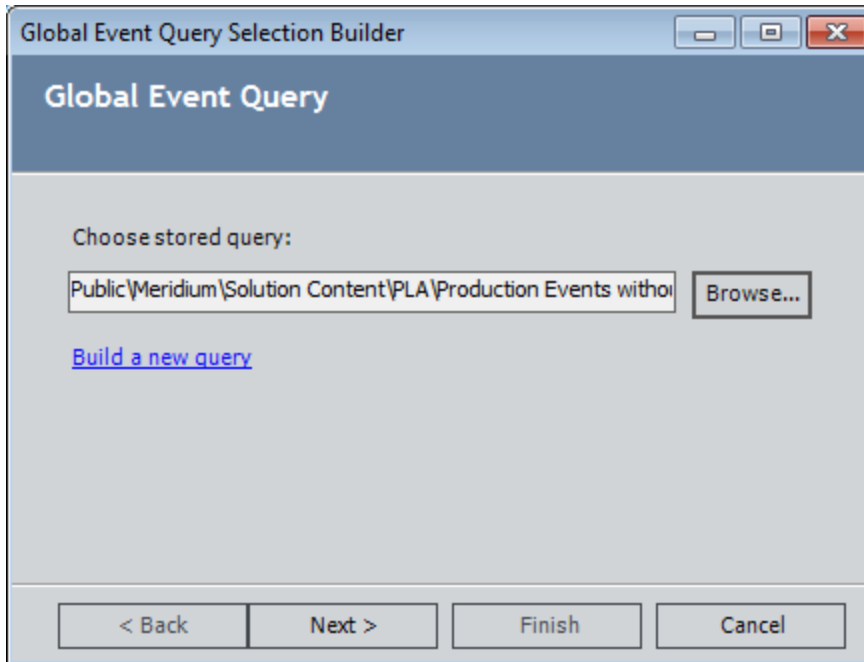
5. Click the **Next** button to confirm that you want to use this query to generate the list of global event records.

    The next screen of the **Global Event Query Selection Builder** appears.
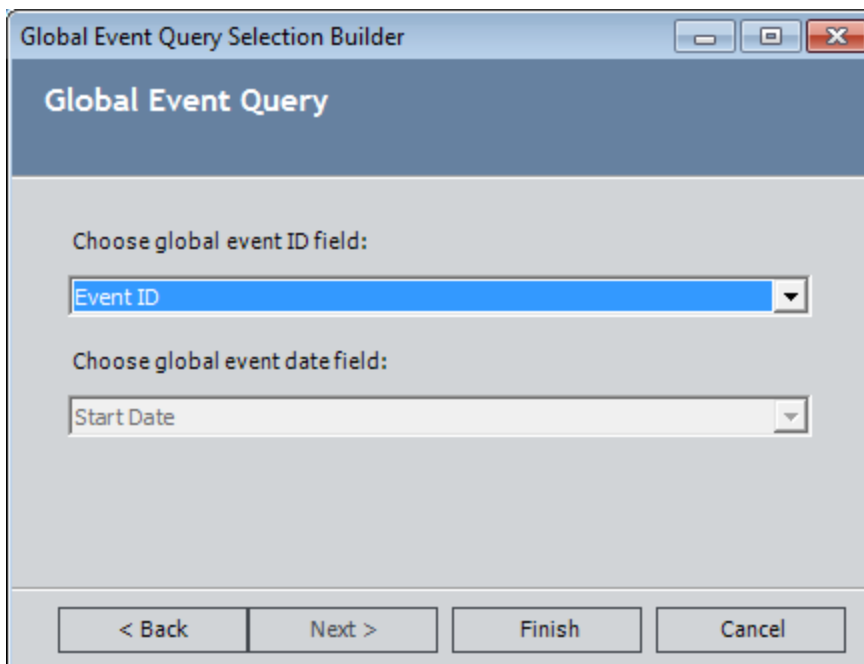
6. In the **Choose global event ID field** list, select the field that will return the value you want to display in the **EventProperties** window for the Event ID. For example, if in the **EventProperties** window you want to see as the Event ID the value stored

in the *Shutdown Type* field, you would select the field that corresponds to the Shutdown Type. Only character fields that exist in the selected query will appear in the list. If the selected query contains only one character field, that field is selected by default, and the **Choose global event ID field** list is disabled.

In the following example, the global event tooltip shows that the value in the Shutdown Type field of the selected global event record is Planned Turnaround.

7. In the **Choose global event date field** list, select the field that will return the value you want to display in the **EventProperties** window for the Event Date. For example, if in the **Event Properties** window you want to see as the Event Date the value stored in the *Event Start Date* field, you would select the field that corresponds to the Event Start Date. Only date fields that exist in the selected query will appear in the list. If the selected query contains only one date field, that field is selected by default, and the **Choose global event date field** list is disabled.

In the following example, the global event tooltip shows that value in the **Event Start Date** field of the selected global event record is **1/30/2005**.

8. Click the **Finish** button.

The list of global event records appears in the **Global Events** dialog box.

9. Select global event records to display on the Timeline by selecting the **Overlay** check boxes for the desired records.

10. Click **OK**.

The Timeline displays the selected records. Each global event record is represented as a green vertical line on the Timeline.

Building a New Query to Generate a List of Global Event Records

To build a new query to generate a list of global event records:

1. In the Meridium APM Framework, on the **Timeline** page, on the **View Options** menu, click the **Overlay Global Events**.

    The most recently selected global event query runs, and the **Global Events** dialog box appears, displaying the records returned by the query.
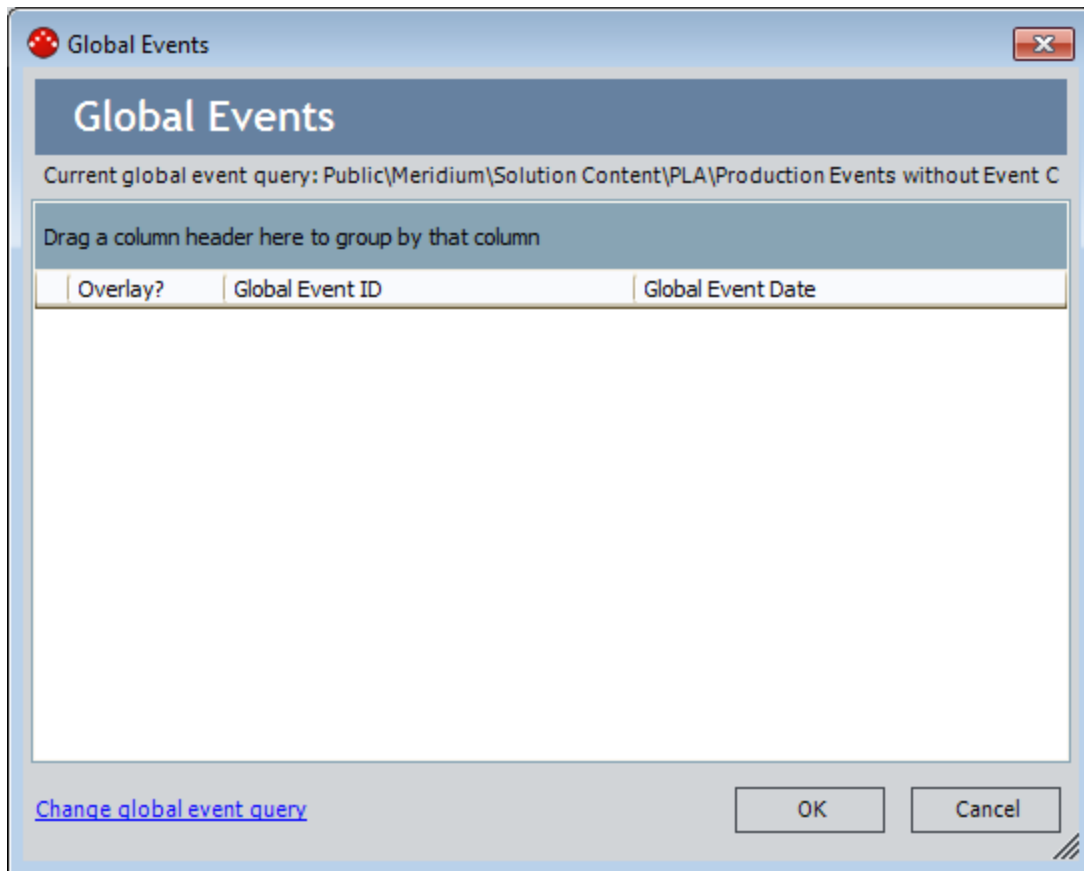
> **Note:** If no records are returned by the current global event query or a query has not yet been selected, you will receive a warning message. Click **OK** to display the **Global Events** dialog box.

2. Click the **Change global event query** link.

    The **Global Event Query Selection Builder** appears.

3. Click the **Build a New Query** link.

   The **Query Builder** appears.
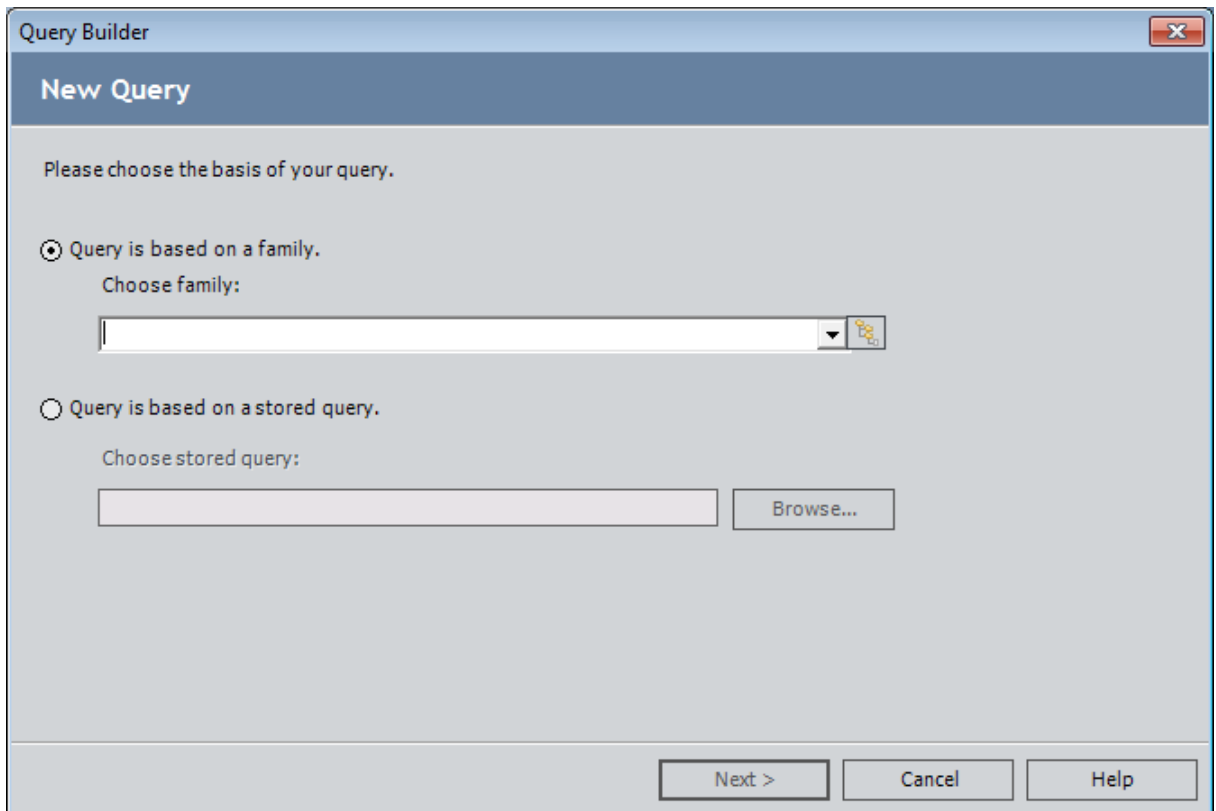
4. Proceed through the Query Builder as desired, and click the Finish button. You can build any query that will return the desired records. Note that the query must contain at least one date field and one character field.

The Save Query dialog box appears.
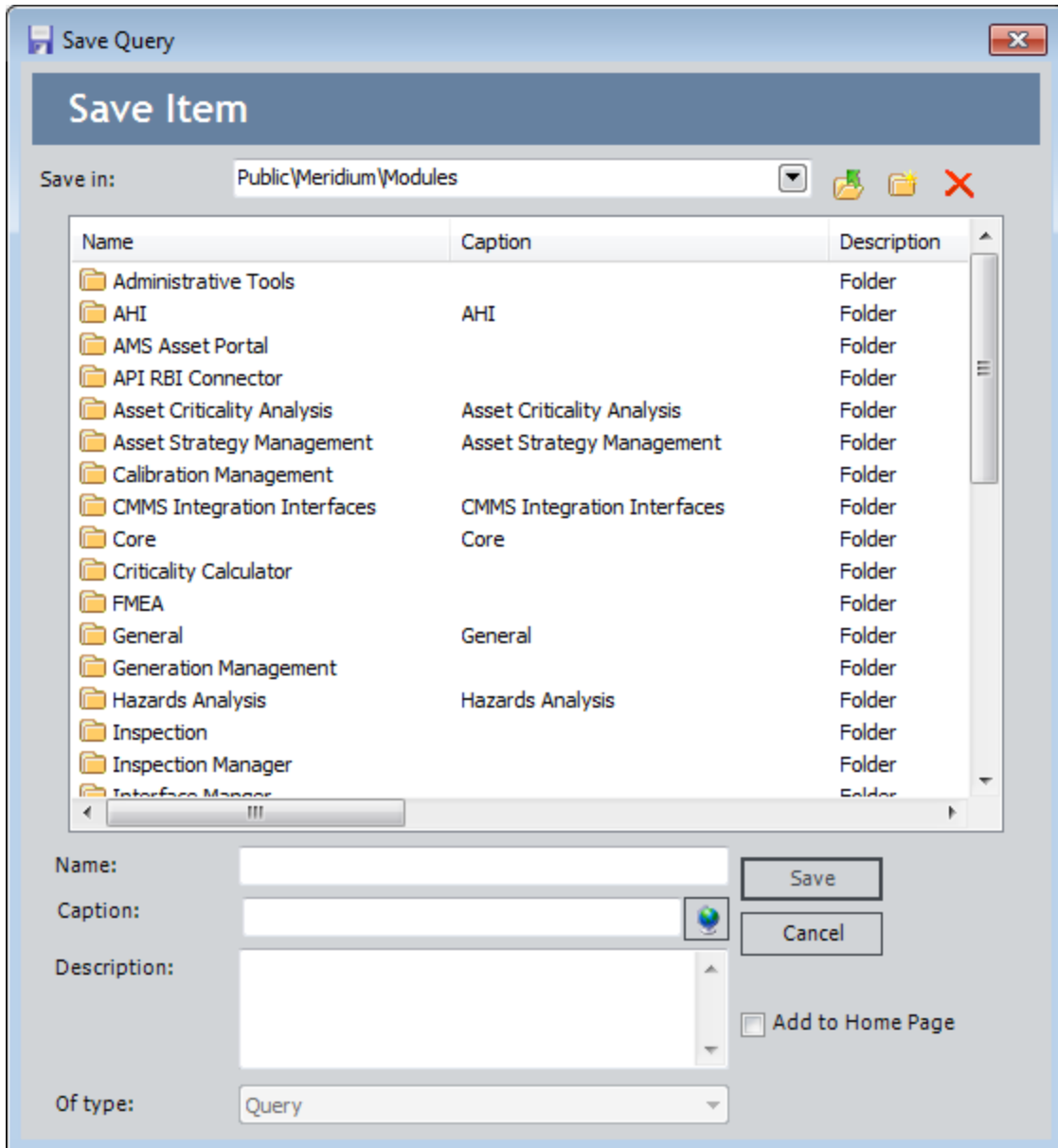
5. Save the query to the desired location.

The next screen of the Global Event Query Selection Builder appears.
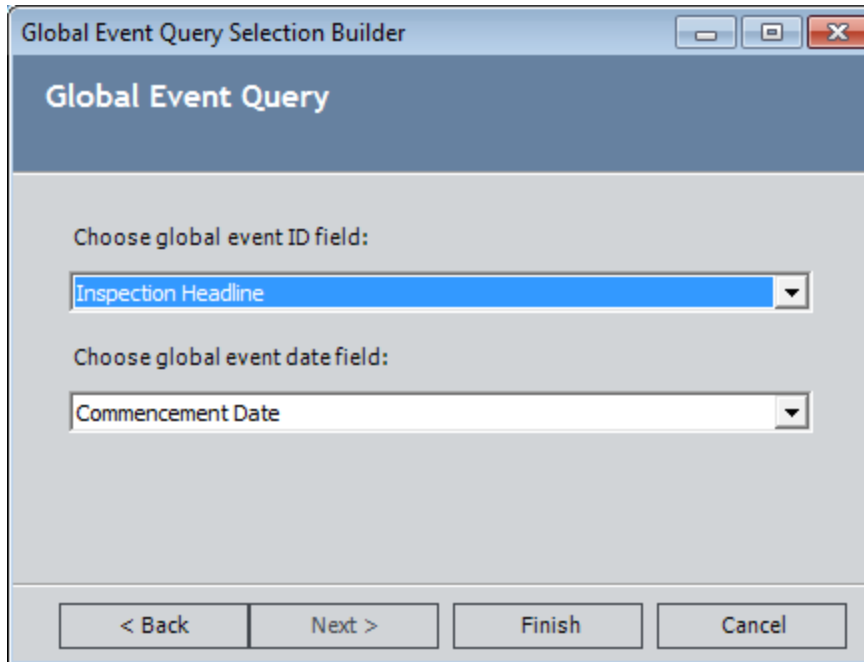
6. In the **Choose global event ID field** list, select the field that will return the Event ID value that you want to display in the pop-up box that displays the Event Properties. For example, if, in the global event tooltip, you want to see as the Event ID the value stored in the *Shutdown Type* field, you would select the field that corresponds to the Shutdown Type. Only character fields that exist in the selected query will appear in the list. If the selected query contains only one character field, that field is selected by default, and the **Choose global event ID field** list is disabled.

   In the following example, the global event tooltip shows that the value in the **Shutdown Type** field of the selected global event record is **Planned Turnaround**.

7. In the **Choose global event date field** list, select the field that will return the Event Date value that you want to display in the pop-up box that displays the Event Properties. For example, if, in the global event tooltip, you want to see as the Event Date the value stored in the *Event Start Date* field, you would select the field that corresponds to the Event Start Date. Only date fields that exist in the selected query will appear in the list. If the selected query contains only one date field, that field is selected by default, and the **Choose global event date field** list is disabled.

   In the following example, the global event tooltip shows that value in the **Event Start Date** field of the selected global event record is **1/30/2005**.

8. Click the **Finish** button to confirm your selection.

    The **Global Events** dialog box appears again. The name of the global event query that you selected is listed on the dialog box, and the list is now populated with global event records based on the selected query.
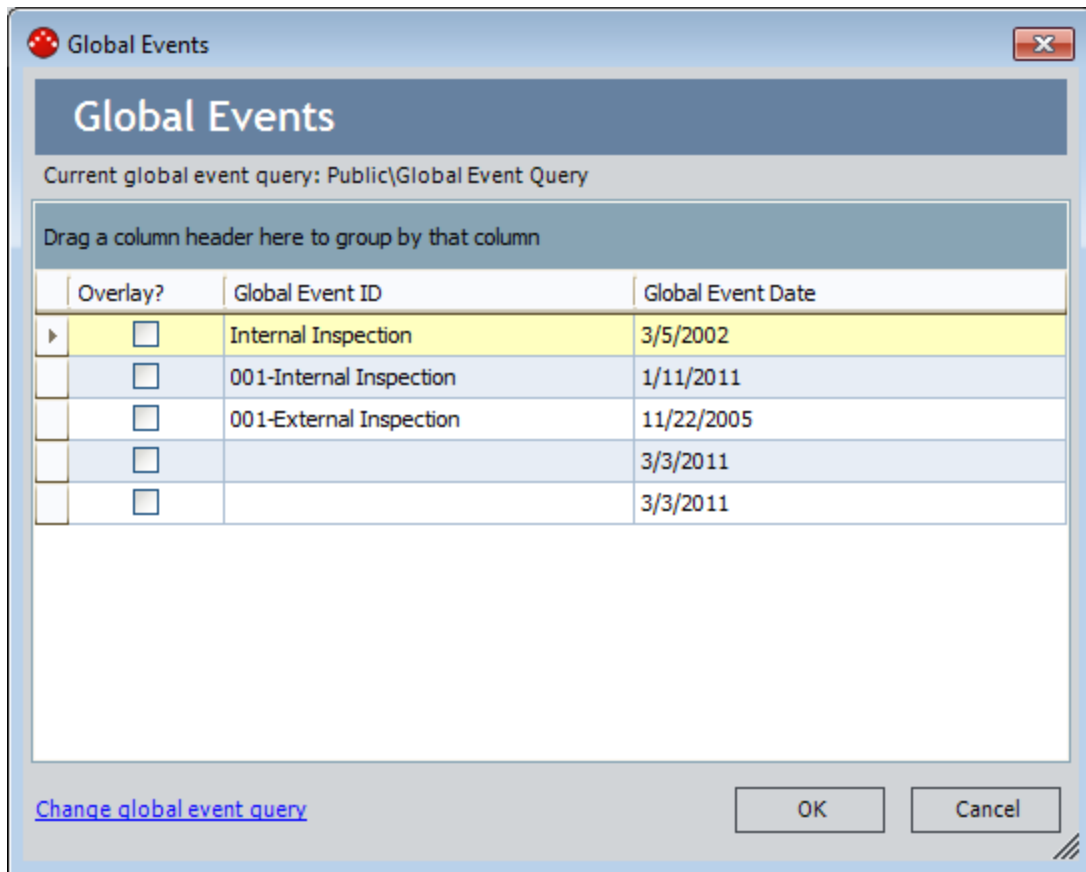
9. Select global event records to display on the Timeline by selecting the **Overlay** check boxes for the desired records.

10. Click **OK**.

    The Timeline displays the selected records. Each global event record is represented as a green vertical line on the Timeline.

About Creating a Link Between Task Records and Global Event Records

The process of matching the Next Date of a Task record to a global Event record date and creating a link between the two records is called *snapping* the Task record to the global Event record. To do this, the entity families must be related through the Has Task relationship family.

For example, you might want to snap a Task record to a Plant Shutdown global Event record. Suppose a Task record indicates that a particular Motor should be replaced on May 1, 2005. A Plant Shutdown record, however, indicates that the facility is going to shut down between April 30, 2005 and May 5, 2005. Because the Motor is scheduled to be replaced when the plant is scheduled to be shut down, you could compare the appropriate dates of the two records, May 1, 2005, and May 5, 2005, and decide to change the implementation date of the Task record to match the End Date of the plan shutdown operation, May 5, 2005. In this case, you would snap the Task record to the Plant Shutdown record, which would create a link between the Plant Shutdown record and the Task record.

You can create a link between a single Task record and a global Event record, or you can create a link between multiple Task records and a global Event record at once.

If you want to snap individual Task records to a global Event record, you can do so while the global Event record line is green. If you want to snap multiple Task records to global Event records, however, you must activate the global Event record line, which will change it to a yellow line.

When you snap a Task record to a global Event record, the diamond representing the Task record will change from blue to green and will lie directly on the line representing the global Event record, as shown in the following image.

Creating a Link Between a Single Task Record and a Global Event Record

While one or more global Event records are displayed on the Timeline, if you move a Task record icon representing a Task record that is not linked to a global Event record over a line representing a global Event record, the system will prompt you to confirm that you want to snap the Task record to the global Event record. After you confirm the action, the value in the Task record's **Next Date** field, which indicates the next date that the action specified in the Task record will be performed, is set to the global Event record date that you specified in the **Global Event Query Selection Builder**. When you save changes to the Timeline, a link will be created between the Task record and the global Event record.

Note that you can snap a Task record to a global Event record only if the specified date of the global Event record falls between the Minimum Date and Maximum Date values of the Task record. For example, in the following image, the line representing the global Event record falls within the Minimum Date value and the Maximum Date value of the Task record.
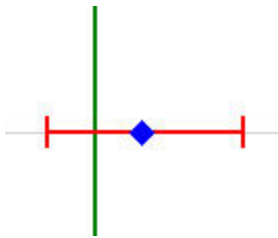
In order to create a link between a Task record and a global Event record, the families to which the two records belong must be related through the Has Task relationship.

To create a link between a single Task record and a global Event record:

1. In the Meridium APM Framework, on the **Timeline** page, drag the blue diamond until it is *exactly* on top of the vertical line that represents the global Event record.

   A message appears, asking if you want to snap the Task record to the global event record.

   > **Note:** If you drag a Task record icon *close to* a global Event line, you will also be prompted to snap the Task record even if the Next Date in the Task record does not reflect the exact date of the global Event record. If you choose to snap the Task record, the Next Date in the Task record does not get updated to the date on the global Event record. To prevent this problem, when dragging a Task record, make sure that the Next Date of the Task record matches *exactly* the date of the global Event record before you release the mouse button and choose to snap the Task record.

2. Click the **Yes** button.

   The Task record is snapped to the global Event record. The diamond symbol changes from blue to green.

   > **Note:** Since a snapped Task record is linked to a global Event record, the action specified in the Task record is presumably executed on the same date as the date of the global Event record. Note, however, that if the global Event date changes, the Meridium APM system will not update the **Next Date** in the associated Task record. The Task record will still be linked to the global Event record. The situation will need to be resolved manually.

3. On the **Common Tasks** menu, click the **Save Changes** link to save your changes.

   The snapped Task record is saved with the updated Next Date, and a link between the Task record and the global Event record is created.

Creating a Link Between Multiple Task Records and a Global Event Record

To create a link between multiple Task records and a global Event record:

1. In the Meridium APM Framework, on the Timeline, pause on the green vertical line representing the global Event.

   A pop-up box appears, displaying the Event Properties. The Event ID and Event Date are determined by the Event ID field and Event Date field you selected in the **Global Event Query Selection Builder**.

2. To activate the global Event record, click the exact spot where the line extending from the tooltip intersects with the green, global Event record line. An Active global Event record is one to and from which Task records can be snapped.

   > **Note:** Only one global Event record can be active at any given time.

   The green line changes to yellow, and a message appears, indicating that the global Event record is now active, and you can snap or unsnap Task records to or from the global Event.

3. Click **OK**.

4. On the **View Options** menu, click the **Snap Tasks** link.

   The **Snap Tasks** dialog box appears, displaying a list of only the Task records that can be snapped to the global Event record.

   > **Note:** If you selected multiple global Event records to display on the Timeline, and one of the global Event records does not fall within the time base, no Task records will appear on the **Snap Tasks** dialog box. To ensure that the correct Task records appear on the **Snap Tasks** dialog box, display only global Event records that fall within the time base.

5. Select the **Snapped** check box for each Task record that you want to snap to the global Event record.

   -or-

   Click the **Select All** button to snap all Task records in the list to the global Event record.

6. Click **OK**.

The Task records that you selected appear on the Timeline as green diamonds and are positioned directly on the global Event line.
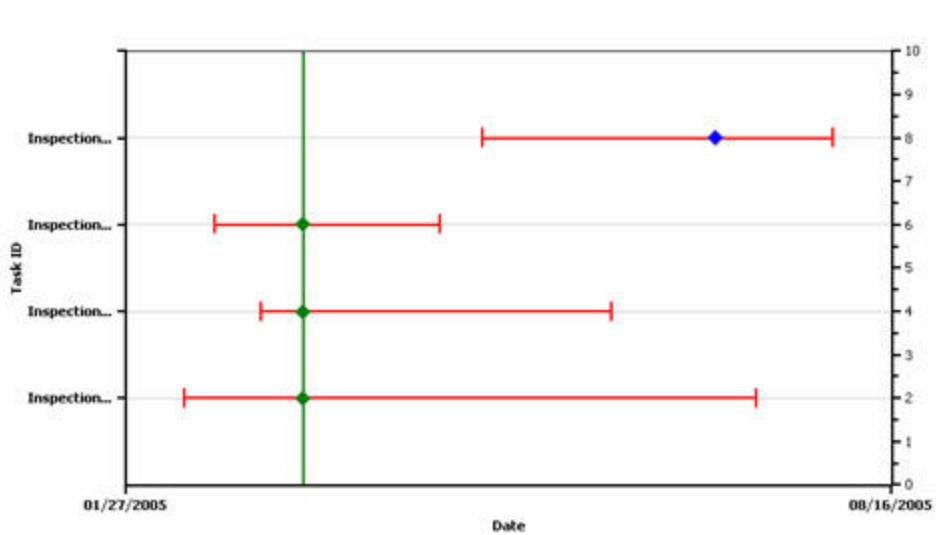
The **Next Date** of the snapped Task records is now the same as the global Event record date you selected in the global Event query selection process. The global Event record is again represented by a green line, indicating that it is not Active.

7. On the **Common Tasks** menu, click the **Save Changes** link to save your changes.

The snapped Task records are saved with the updated Next Date, and a link between the Task records and the global Event record is created.

Changing the Next Date of Task Records

To change the Next Date of a Task record on the Timeline:

1. In the Meridium APM Framework, on the **Timeline** page, select a Task record by clicking the diamond symbol representing the Next Date. You can select a Task record that is or is not linked to a global Event record.

2. Drag the icon across the Timeline.

   If you drag an icon representing a Task record that is linked to a global Event record, when you release the mouse button, a message will appear, asking if you want to unsnap the Task record from the global Event record. Click **OK** to confirm the action.

   -or-

   If you drag an icon representing a Task record that is not linked to a global Event record, as you drag the icon, the **Next Date** (the next date on which the action specified in the Task record will be performed) will change on the tooltip.

   In the following image, the Task record symbol has not yet been moved, and the Next Date is 8/7/2005.
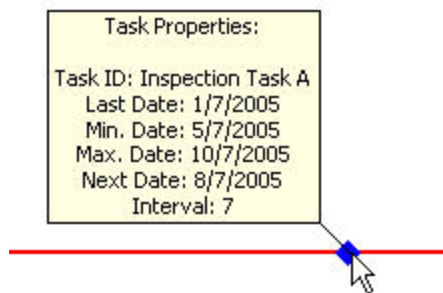
   After moving the Task record symbol across the Timeline, the Next Date is updated to reflect the move. In the following image, the Task record was moved so that the Next Date would be 8/19/2005.
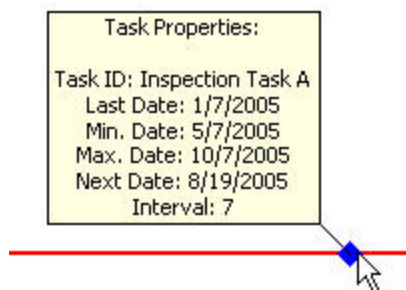
3. On the **Common Tasks** menu, click the **Save Changes** link to save the changes to the Task record.

Resetting the Timeline

Resetting the Timeline will undo all of the changes that you made to the Task record dates on the Timeline.

To reset the Timeline:

- In the Meridium APM Framework, on the **Timeline** page, on the **View Options** menu, click the **Reset Timeline** link.

  All changes to the Timeline that have not been saved are discarded. All Task record dates are returned to the original settings. If you displayed global Event records on the Timeline, filtered the display of the Task records on the y-axis, or changed the time base, all of these changes will be discarded.

  > **Note:** The same global Event query remains selected and the same list of global Event records is available when you click the **Overlay Global Events** link, but any global Event records that you chose to display on the Timeline are removed from the Timeline.

Saving Changes to the Timeline

To save changes to the Timeline:

- In the Meridium APM Framework, on the **Timeline** page, on the **Common Tasks** menu, click the **Save Changes** link.

  Any changes you made to Task record dates are saved to the database.

Dashboards Catalog Folder Structure

Information on the Catalog folders that store baseline dashboards and their underlying queries can be found in the module or work process documentation for the relevant baseline dashboard. Like all Catalog content, the Meridium APM database includes a copy of each dashboard and its underlying queries in the Baseline folder and a corresponding Public folder.

Task Management Catalog Folder Structure

The main Task Management Catalog folder is \\Public\Meridium\Modules\Task Management, which contains baseline queries that are used by default in Task Management.

If desired, you can store additional items in this folder or create new subfolders to store additional items.

> **Note:** All Public folders exist by default. Any baseline item in the Public folder is also available in the corresponding Baseline folder. Throughout this documentation, however, we refer to items in the Public folder.

The Catalog folder \\Public\Meridium\Modules\Task Management contains the queries listed in the following table. In this table, the values listed in the Query column refer to the query names. The baseline query captions are the same as the query names.

Query	Behavior and Usage
Available Tasks	After displaying a prompt asking for an Asset ID and Task Family, returns all Task records that belong to the selected Task family and contain the selected value in the Asset ID field. Because this query is specified as the default Task query on the **Task Management Administrative Tasks** page, this query is run when you access the **Task List** page.
Recommendations by Asset	This query is run when you click the **Show Recommendations** link on the **Asset Options** menu on the **Task List** page. The query returns a list of all Recommendation records that are linked to the Equipment or Functional Location record that is linked to the Task record that was selected on the **Task List** page. Note that this query is not meant to be run on its own. In addition, you should not change the name of this query or move it to another Catalog folder.
Recommendations by Specific Equipment	This query is no longer used by the baseline Task Management feature. It was provided in previous releases to facilitate a workflow that is now accomplished using the Recommendation by Asset query.

Task

The following table provides a list and description of some of the fields that exist in Task records and are available on the baseline Task datasheet. The information in the table reflects the baseline state and behavior of these fields. If your implementation has been customized, these fields may behave differently, and fields in addition to those listed here may be available.

> **Note:** The fields are listed in the order in which they appear on the Task datasheet.

Field	Data Type	Description	Behavior and Usage	Notes
Desired Interval	Number	A number indicating how often the task should be completed.	The value in the Desired Interval field must be less than or equal to the value in the Maximum Interval field.	Before you can save a value in this field, the Last Date field must contain a value. The next date is calculated automatically using both the last date and desired interval. Note that this rule does *not* apply to Inspection Task records.
Desired Interval UOM	Character	The unit of measure associated with the value in the Desired Interval field.	On the datasheet, the field displays a list with the following values: • Days • Months • Weeks • Years The value is set by default to *Months*. In the baseline database, you can change the value in root Task records only.	None

Field	Data Type	Description	Behavior and Usage	Notes
Last Date	Date	The last date that the task was executed.	When you execute a Task record, you are prompted for an execution date, which will be used to populate the Last Date field in the Task record and the Date Executed field in the Task Execution record to which it is linked. In Inspection Task records, the Last Date field can be updated automatically by executing the MI_InspectionTasksUpdate strategy rule.	None
Max Date	Date	The latest date on which the task should be completed.	This value is calculated automatically based upon the values in the Max Interval and Max Interval UOM fields.	None
Max Interval	Number	The maximum interval at which the task should be completed.		None
Max Interval UOM	Character	The unit of measure associated with the maximum interval.	This value in this field is set automatically to the value in the Desired Interval UOM field and cannot be modified.	None

Field	Data Type	Description	Behavior and Usage	Notes
Min Date	Date	The earliest date on which the task should be completed.	This value is calculated automatically based upon the values in the Min Interval and Min Interval UOM fields.	None
Min Interval	Number	The minimum interval at which the task should be completed.	You can type a value directly in the **Min Interval** cell.	None
Min Interval UOM	Character	The unit of measure associated with the minimum interval.	This value in this field is set automatically to the value in the Desired Interval UOM field and cannot be modified.	None

Field	Data Type	Description	Behavior and Usage	Notes

Modifiable	Logical	A value that indicates whether or not the remaining fields should be disabled.	On the datasheet, this field appears as a check box. The state of the check box determines the behavior of the remaining fields, depending upon the family to which the Task record belongs.	• In Inspection Task records, the Modifiable field behaves differently than it does in root Task records. In Inspection Task records, the **Modifiable** check box is selected when you create the record but cleared automatically after the record is saved.
				
				• In Calibration Task records, the **Modifiable** check box is cleared by default. When it is cleared, all remaining fields become disabled with the exception of the following fields:
				
				  ▪ Task State
				
				  ▪ Task Assigned To
				
				  ▪ Route Number
				
				  ▪ Reoccurring
				
				If you modify the value in any of the fields in the preceding list, the **Modifiable**

Field	Data Type	Description	Behavior and Usage	Notes
Next Date	Date	The next date on which the task should be executed.	The value is calculated based upon the values in the Last Date and Desired Interval fields.	None
Rejectable	Logical	A value that indicates whether or not the Task record can be rejected.	On the datasheet, this field displays a check box. If the check box is cleared, you cannot reject the Task record. If it is selected, you can reject the Task record.	In Calibration Task records, the **Rejectable** check box is cleared by default.
Reoccuring	Logical	A value that indicates whether or not the task should occur on a regular schedule.	On the datasheet, this field appears as a check box. If the check box is *clear*, when you execute the Task record, the state will be changed to *Completed*. If the check box is *selected*, when you execute the Task record, the state will be changed to *Scheduled*.	None
Task Description	Text	A longer description of the task that should be completed.	You can type the description directly in the **Task Description** cell.	You can use this field to store any additional information about the Task record that is not stored in another field.
Task ID	Character	A value that identifies the task.	You can type the Task ID directly in the **Task ID** cell.	You must enter a value in this field when you create a Task record.

Field	Data Type	Description	Behavior and Usage	Notes
Task Type	Char-acter	The type of task that should be com-pleted.	On the datasheet, the **Task Type** list is pop-ulated with the Task Type IDs of all the [records that belong to the Task Types family]().	None
Uncon-strain Min/Max Dates	Logical	A value that indic-ates how the Min Date and Max Date fields will be pop-ulated.	When the check box is cleared, the Min Date and Max Date fields will be pop-ulated automatically based on the Last Date, Min Interval, and Max Interval val-ues. If you select this check box, the Min Date and Max Date fields will be cleared. Because these fields are disabled, how-ever, you will be unable to enter a Min Date or Max Date.	None

About Query Performance

Query performance is influenced by a number of factors, including hardware efficiency (e.g., how fast the Meridium APM client and Server machines are), the system's workload at the time you run the query (e.g., how many users are logged in and making requests), and the efficiency of the database (e.g., how well the database is being maintained). As you can see, many of the factors that influence query performance are beyond your control. For example, you cannot control how many other users are currently using the system.

Query performance is also affected by a number of factors that you can control, including how you construct the query and the options that you choose for running it. This section of the documentation discusses things that you can do to maximize the performance of queries.

> **Note:** The suggestions provided here may not improve the performance of all queries. In addition, other factors (e.g., deficiencies in hardware resources) may negate any improvement that would otherwise be achieved by implementing these suggestions.

Limiting the Size of the Query

Generally, the more data that you attempt to retrieve with a query, the longer the query will take to run. For example, a query that retrieves data for all fields in the Recommendation family (which has many subfamilies) will take longer to run than a query that returns only a few fields from the Meridium General Recommendation family (a subfamily of the Recommendation family).

Knowing this, you can improve query performance by limiting the amount of data that you return in the query results. For example:

- Always query on the lowest-level family possible. For example, instead of querying on a family, query on individual subfamilies. If you need to query on more than one subfamily, you might want to create more than one query.

- Only include in your query results the fields that you actually need. Instead of returning the information for ALL fields in a given family, limit your results to only the specific fields that you want to analyze.

- Use criteria to limit results to a specific set of records. For example, if you are interested only in the Pumps associated with a given manufacturer, limit your query results by applying criteria to the Manufacturer field.

Note: Performance improvement from limiting the size of the query will be seen only when comparing queries that are otherwise similar.

Running Select Queries in Unformatted Mode

Meridium APM provides two output modes for running Select queries:

- **Formatted Mode:** Formats the output based upon format and valid values rules defined for the family.
- **Unformatted Mode:** Returns values exactly as they exist in the database, with no additional formatting applied.

Because unformatted mode returns results without taking the extra step of applying formatting before displaying results, running a query in unformatted mode can be more efficient than running the same query in formatted mode. This performance improvement is particularly significant when you run a query that includes many formatted fields.

If you find that a query seems to be running slowly, check the output mode. If it is set to Formatted Mode, change it to unformatted mode, and run the query again.

Note: Unformatted mode is the default output format for new Select queries.

System Code Tables Used by Strategy Rules

The following System Code Table is used by Strategy Rules.

Table ID	Table Description
MI_STRATEGY_REFERENCE	CORE, Strategy references.

System Code Tables Used by Task Management

The following System Code Table is used in Task Management.

Table ID	Table Description
MI_TASK_TYPE_REFERENCE	CORE, Reference values to limit task type lists when creating Task records.

This System Code Table is used to determine the list of values that appear in the **Reference** list when creating a new Task Type record in the Meridium APM Framework application. Meridium APM provides System Code IDs of INSPECTION, CALIBRATION, and RCM, so the **Reference** list is automatically populated with these three references on the Task Type datasheet.

> **Note:** We recommend that you not change the MI_TASK_TYPE_REFERENCE System Code Table. If you do, the Meridium APM Framework application will not function as designed. Specifically, the **Task Types** list on the Task datasheet will not be filtered for Inspection and Calibration tasks.

Parameters for the Dashboard URL

The URL for the **Dashboard** page, **meridium://WEBBROWSWER**, accepts the parameters listed in the following table.

Parameter Name	Description	Accepted Value	Notes
RelativeUrl	Specifies that the dashboard identified by the **Path** parameter will be displayed on the **Dashboard** page.	Meridium/Dashboard/Index.html	This parameter is required.
Path	Identifies the Catalog location of the dashboard that you want to view on the **Dashboard** page.	The Catalog location of the dashboard	This parameter is required.

Examples of the Dashboard URL

- meridium://WEBBROWSER?RelativeUrl=Meridium/Dashboard/Index.html?**Path-h=Public\Meridium\Modules\Mechanical Integrity\Dashboards\Mechanical Integrity Dashboard**

  Opens the baseline Mechanical Integrity dashboard on the **Dashboard** page.

- meridium://WEBBROWSER?RelativeUrl=Meridium/Dashboard/Index.html?**Path-h=Public\Meridium\Modules\Operator Rounds\Dashboards\OPR Dashboard**

  Opens the baseline Operator Rounds dashboard on the **Dashboard** page.

Parameters for the Dataset URL

The URL for the Dataset Editor, **meridium://Registry/Dataset**, accepts the optional parameters listed in the following table. Note that a link created from the path with NO parameters will launch an empty **Dataset Editor** page. For example of how you can combine these parameters to construct URLs, [click here](#).

Parameter Name	Description	Accepted Value(s)	Notes
NewDataset	Launches the **Dataset Builder**, where you can create a new dataset.	True	This parameter also accepts False as a value. Note, however, that since the only reason for using this parameter is to create a new dataset, you should *never* need to set this parameter to False.
Key	Loads the dataset with the specified key value.	A numeric dataset key.	Use this parameter to open an existing dataset.
Path	Loads the dataset stored in the specified location.	The file name and path of the dataset that you want to open.	Use this parameter to open a dataset that has been saved to the Catalog.

Examples of the Dataset URL

- meridium://Registry/Dataset?**NewDataset=True**

  Opens the **Dataset Editor** page and launches the **Dataset Builder**, where you can begin the process of creating a new dataset.

- meridium://Registry/Dataset?**Key=1234567**

  Opens the **Dataset Editor** page and loads a saved dataset with the Key 1234567.

- meridium://Registry/Dataset?**Path-h=Public\Meridium\Modules\Calibration\CalibrationDataset**

  Opens the dataset called CalibrationDataset, which is stored in the Catalog folder \\Public\Meridium\Modules\Calibration.

Parameters for Graph URLs

The URL for graphs, **meridium://Registry/GraphViewer**, accepts the parameters listed in the following table. Note that a link created from the path with NO parameters will generate an error. You must specify either to create a new graph or open an existing graph. For example of how you can combine these parameters to construct URLs, [click here](#).

Parameter Name	Description	Accepted Value(s)	Notes
DatasetKey	Displays the **Graph Setup** window, where you can choose options for creating a graph based on the specified dataset.	The key value of the dataset that you want to use for creating the new graph.	To use this parameter for creating a new graph, you must use NewGraph=True.
DatasetPath	Displays the **Graph Setup** window, where you can choose options for creating a graph based on the specified dataset.	The Catalog path and name of the dataset that you want to use for creating the new graph.	To use this parameter for creating a new graph, you must use NewGraph=True.
GraphPath	Opens the Graph Viewer, displaying the graph specified by the GraphPath parameter.	The Catalog path and name of the graph that you want to open.	If the graph is based on a query that contains a prompt, you can [pass values in for the prompts](#) via the URL.
GraphKey	Opens the Graph Viewer, displaying the graph specified by the GraphKey parameter.	The key value of the graph that you want to open.	If the graph is based on a query that contains a prompt, you can [pass values in for the prompts](#) via the URL.

Parameter Name	Description	Accepted Value(s)	Notes
GraphTitle	Specifies the title to display for the graph.	The desired alphanumeric title.	This parameter is optional. If you omit this parameter, the Catalog item name of the saved graph will serve as the title of the graph.
NewGraph	Launches the **Choose a Query or Dataset for Graph** dialog box, where you can select a saved query or dataset from which to create a new graph.	True	This parameter also accepts False as a value. Note, however, that since the only reason for using this parameter is to create a new graph, you should *never* need to set this parameter to False.
PromptHeader	Runs the underlying query and displays the prompt values in a list at the top of the graph. A **Run Query** button is also displayed so that the user can select a different value for the prompt(s) and re-run the query to modify the results displayed on the graph.	1	None
QueryKey	Executes the query specified by the key and displays the **Graph Setup** window, which you can use to select options for creating the graph.	The key value of the query that you want to use for creating the new graph.	To use this parameter for creating a new graph, you must use NewGraph=True.

Parameter Name	Description	Accepted Value(s)	Notes
QueryPath	Executes the specified query and displays the **Graph Setup** window, which you can use to select options for creating the graph.	The Catalog path and name of the query that you want to use for creating the new graph.	To use this parameter for creating a new graph, you must use NewGraph=True.

Examples of Graph URLs

- meridium://Registry/GraphViewer?**NewGraph=True**

  Opens the **Create New Graph** dialog box, where you can choose to create a new graph based on a query, a dataset, or an external template.

- meridium://Registry/GraphViewer?**GraphPath-h=Public\Meridium\Graphs\SavedGraph**

  Opens the graph SavedGraph in the Graph Viewer.

- meridium://Registry/GraphViewer?GraphPath=Public\Meridium\Graphs\SavedGraph&**GraphTitle=Unique Title for Saved Graph**

  Opens the graph SavedGraph in the Graph Viewer. Instead of displaying *SavedGraph* as the title, *Unique Title for Saved Graph* is displayed.

- meridium://Registry/GraphViewer?GraphPath=Public\Meridium\Graphs\SavedGraph&**PromptHeader=1**

  Opens the graph SavedGraph in the Graph Viewer. The prompts included in the underlying query are displayed above the graph. You can select any prompt value and click the **Run Query** button to refresh the graph based on the updated prompt value.

- meridium://Registry/GraphViewer?**QueryPath-h=Public\Meridium\Queries\PumpQuery**

  Creates a new graph using the query PumpQuery, which is stored in the Catalog folder \\Public\Meridium\Queries.

- meridium://Registry/GraphViewer?**DatasetKey=1234567**

  Creates a new graph using the dataset that is identified by the key value 1234567.

Parameters for Query URLs

> **Note:** This topic provides details on URLs that will open existing queries or construct new queries. It does not provide details on constructing hyperlinks *within* queries. To construct a hyperlink within a query, you can use a URL to any external location or any internal URL discussed in this documentation.

The URL for queries, **meridium://Registry/QueryBuilder**, accepts the parameters listed in the following table. Note that a link created from the path with NO parameters will display the **New Query** dialog box, where you can choose to create a new query using the Query Builder or the query design. For example of how you can combine these parameters to construct URLs, click here.

Parameter Name	Description	Accepted Value(s)	Notes
p0, p1, p2, etc.	Specifies values that you want to pass in to the prompts that exist in the query.	Any value that is acceptable for the prompt type (e.g., numeric values for numeric prompts).	You can pass prompt values in to the query only when the query contains prompts. Note that while *prompts* are specific to queries, prompt parameters are not specific to the query URL and are, therefore, not discussed in detail in this section. Query prompt values can be passed into a query via the URL whenever the URL accepts a query as a parameter value. For example, you can use a graph URL to open an existing graph. If the graph that you choose to open is based upon a query that contains prompts, you can pass values in to those prompts via a graph URL.

Parameter Name	Description	Accepted Value(s)	Notes
PromptHeader	Runs the specified query and displays the prompt values in a list at the top of the results. A **Run Query** and **Stop** button are also displayed so that the user can select a different value for the prompt(s) and rerun the query.	1 2	The PromptHeader parameter can be used only when RunOnly=1. When PromptHeader=1, the query is run and the results are displayed automatically. When PromptHeader=2, the query is not run automatically. Instead, the prompts appear so that you can specify prompt values before running the query manually.
QueryPath	Opens the specified query.	The Catalog path and name of the query you want to open.	In the Catalog path and name, you must use the same capitalization that exists in the Catalog. If you specify the path to a query that contains prompts, you can pass values in to the query prompts via the URL.
RunOnly	Runs the specified query in *view-only* mode, which displays the query results but hides the **Design View**, **SQL View**, **Source**, and **Options** links on the **Query Tasks** menu, thus preventing the user from modifying the query.	1	This parameter should be used with the QueryPath parameter, which lets you specify which query you want to open.

Parameter Name	Description	Accepted Value(s)	Notes
StartMode	Opens a query in the specified mode.	design sqledit build run	*These parameter values are case sensitive. You must use all lowercase.* Use this parameter in conjunction with the QueryPath parameter, which specifies which query you want to open. • **design:** The specified query will open and display the query design. • **sqledit:** The specified query will open and display the SQL code. • **build:**The system will execute the specified query and then immediately display the **New Query** dialog box so that you can create a new query. • **run:** The query will be executed, and the results will be displayed. Note that if you simply want to run a query and display the results, you can omit the StartMode parameter altogether and define just the QueryPath parameter.

Examples of Query URLs

> **Note:** The following examples show how to construct URLs for *accessing* queries. These may or may not be the types of URLs that you want to use for adding URLs to queries themselves.

- meridium://Registry/QueryBuilder

  Displays the **New Query** dialog box, where you can choose to create a new query using the **Query Builder** or via the query design.

- meridium://Registry/QueryBuilder?**QueryPath=Public\Meridium\Queries\Work Orders**

  Opens the Work Orders query, which is stored in the Catalog folder \\Public\Meridium\Queries.

- meridium://Registry/QueryBuilder?QueryPath=Public\Meridium\Queries\Work Orders&**StartMode=design**

  Opens the Work Orders query and displays the query design.

- meridium://Registry/QueryBuilder?QueryPath=Public\Meridium\Reports\Work Orders&**p0=Completed&p1=Pump**

  Assuming that the Work Orders query contains two [prompts](#), the first on status and the second on equipment type, this URL will execute the Work Orders query and pass in *Completed* as the status and *Pump* as the equipment type, returning only the completed work orders that are associated with pumps.

- meridium://Registry/QueryBuilder?QueryPath=Public\Meridium\Reports\Work Orders&p0=Completed,**In Progress**

  Assuming that the Work Orders query contains one [prompt](#) that allows you to select multiple values, this URL will execute the Work Orders query and pass in *Completed* and *In Progress* as the status, returning only completed and in progress work orders.

- meridium://Registry/QueryBuilder?QueryPath=Public\Meridium\Queries\Work Orders&**RunOnly=1**

  Executes the Work Orders query and displays the results. Only the **Run Query** link appears on the **Query Tasks** menu; the **Design View**, **SQL View**, **Source**, and **Options** links are not displayed.

- meridium://Registry/QueryBuilder?QueryPath=Public\Meridium\Queries\Work Orders&RunOnly=1&**promptheader=1**

  Executes the Work Orders query and displays the results. The available prompt values are displayed at the top of the results along with a **Run Query** and **Stop** button. The **Query Tasks** menu is hidden.

- meridium://Registry/QueryBuilder?QueryPath=Public\Meridium\Queries**Heat Exchangers&RunOnly=1&promptheader=1&p0=HX1052**

  Executes the Heat Exchangers query and displays the results. The available prompt values are displayed at the top of the results along with a **Run Query** and **Stop** button. The **Query Tasks** menu is hidden.

  If the query *contains only one prompt*, since the prompt value is supplied via the URL, the query runs automatically and the results are displayed based on the specified prompt value.

  If, however, the query *contains more than one prompt*, since only one prompt value is supplied via the URL, the query will not run automatically. The supplied prompt value will be displayed in the prompt at the top of the results, but the prompt whose value you did not supply will be empty. You must click the **Run Query** button to run the query and display the results.

Syntax for Variable Parameter Values in Query Result Links

When you add a URL to a query, for the parameter values you can pass in any data from the records that are returned by the query. For example, if your query selects Asset records from the database, you could pass data from the Asset records to the URL parameter values. In this case, you will use a *$* (dollar sign) to specify that the parameter value is variable.

For example, consider a query that returns a list of Pumps in the database. You could construct a URL for the Asset ID column of the query so that in the query results, a link would appear to [open each record in the Record Manager](). In this case, instead of specifying a particular Entity Key in the URL to create a static link that would open the same record every time, you would need to specify a variable parameter value so that the URL would be populated with the appropriate Entity Key, depending upon which specific results were returned by the query. The URL that you would add to the Asset ID column in the query would look like this:

meridium://Registry/RecordManager?EntityKey=**$EntityKey$**&FamilyKey=**$FamilyKey$**

This URL contains two variable parameter values, which are shown in blue:

- **$EntityKey$**, which will be replaced with an actual Entity Key in the link.
- **$FamilyKey$**, which will be replaced with an actual Family Key in the link.

Note that each variable value is denoted by dollar signs that surround the variable name. Typically, the variable name will be the Alias for a field in the query but may vary depending on how the query has been configured. In this example, the variable names *Entity Key* and *Family Key* correspond to the Aliases for system fields and will be replaced with actual Entity Keys and Family Keys when the URLs are constructed at run time. You will need to choose a variable that is supported and is appropriate for the URL parameter.

> **Note:** Whenever you pass in a value from a field that is included in the query results, you can use the Alias between the dollar signs, provided that an Alias has been defined. If an Alias is not defined, you can pass in the field ID. We recommend, however, that you define Aliases for every field that will be included in the results. Because Aliases must be unique, this helps eliminate ambiguity in cases where a query uses multiple tables and contains more than one hyperlink, each of which accepts similar values from different tables.

While our example shows how you can pass variable Entity Keys and Family Keys in as URL parameter values, note that you can pass in ANY value that is included in the query and is appropriate for the URL. For example, if your query includes a URL that uses the *AssetID* parameter, you could pass in a variable Asset ID as the parameter value. You might also pass values from the query results to query prompts in order to [link from one query to another]().

Note the following about passing variable values in to hyperlinks in queries:

- When passing in a non-system field as a variable, you must surround the field ID with square brackets. For example, to pass in the value from the Asset ID field, you would need to type *$[Asset_ID_CHR]$* as the variable parameter.

- When running a query in *Unformatted* mode, you must include in the output ANY field that you want to use as a URL parameter value. This rule applies both to system and non-system fields. In our example, to run our query in Unformatted mode, we would need to include **FMLY_KEY** and **ENTY_KEY** as columns in the query and select the **Display** check box to display each column in the output.

- When running a query in *Formatted* mode, you can pass *system* fields in as parameter values *without* including them in the query results. Non-system fields, however, must *always* be included in the query results in order to be used as hyperlink parameter values.

> **Note:** If you omit a system field from the results and still want to pass the values in to a URL parameter, use the field ID instead of the Alias within the dollar signs (e.g., $ENTY_KEY$).

- When your query contains multiple tables and you want to use a hyperlink that accepts one or more parameter values, the Meridium APM system will need to know what the source table is for each parameter value.

  - For fields included in the query results, you can use the Alias. Since each Alias must be unique, the Meridium APM sytem will be able to determine the source table from the field's Alias.

  - For system fields not included in the query results, use one of the following syntax options, depending upon whether or not a table Alias has been defined:

    - **$[TableCaption].FieldID$** where **TableCaption** is the default caption of the table (if a table Alias has not been defined).

    - **$TableAlias.FieldID$** where **TableAlias** is the Alias of the table (if a table Alias has been defined).

Specifying Values for Query Prompts

In many cases, you will have the option of specifying, either directly or indirectly, that a query should be executed when a link is used. For example, you can use the Report Viewer URL to create a link that will open a new report based on a query:

meridium://Registry/ReportViewer?QueryPath=Public\Meridium\Queries\PumpQuery

This URL shows an example of a query being called *directly* by a URL. When a user clicks the link, the system will execute the query *PumpQuery* and use it to create a new report.

Alternatively, you can use the Report Viewer URL to open an existing report:

meridi-um://Registry/ReportViewer?ReportPath=Public\Meridium\Reports\PumpReport

This URL shows an example of a query being called *indirectly* by a URL. In this example, when a user clicks the link, the system will open the report *PumpReport*. If the report is based on a query, that query will be executed.

In either case, whenever a URL will execute a query, if that query contains one or more prompts, you have two options for each prompt:

- You can specify NO value for the prompt, which will cause the prompt dialog box to appear so that the user can select a value.

  -or-

- You can pass values in for the prompt via the URL.

To pass in a value for a query prompt via a URL, use the syntax **<prompt>=<value>**, where:

- **<prompt>** is the prompt ID of the prompt, as defined in the query. Alternatively, you can use the text **Pn**, where **n** is the number of the prompt in the query. Note that prompts are numbered beginning with zero (e.g., 0,1,2,3). For example, to specify a value for the *second* prompt in a query, the parameter would be **P1**.
- **<value>** is the value that you want to supply to the prompt.

So, continuing with the example where we constructed a URL to open the report *PumpReport*, if the query contained two prompts, the first for status and the second for manufacturer, we might modify the URL in the following way:

meridium://Registry/ReportViewer/Report?Path=Public\Meridium\Reports\PumpReport&P0=Active&P1=United Pumps

This modification would automatically select *Active* for the status prompt and United Pumps for the manufacturer prompt.

If a query contains a prompt that accepts multiple values, you can specify multiple values in the URL by using the syntax **<prompt>=<value>,<value>**, where each value is separated by a comma.

Note: If a query contains multiple prompts, and you pass in values for only *some* of the prompts, when you execute the query via the URL, you will be prompted to supply values for ALL the prompts.

Parameters for Report URLs

The URL for the reports, **meridium://Registry/ReportViewer**, accepts the parameters listed in the following table. Note that a link created from the path with NO parameters will generate an error. You must specify either to create a new report or open an existing report. For example of how you can combine these parameters to construct URLs, click here.

Parameter Name	Description	Accepted Value(s)	Notes
DatasetKey	Creates a new report using the dataset specified by the key.	The key value of the dataset that you want to use for creating the new report.	To use this parameter for creating a new report, you must use NewReport=True. To create a new report, you must have the full version of the SQL Server Business Intelligence Development Studio installed.
DatasetPath	Creates a new report using the dataset specified by the path.	The Catalog path and name of the dataset that you want to use for creating the new report.	To use this parameter for creating a new report, you must use NewReport=True. To create a new report, you must have the full version of the SQL Server Business Intelligence Development Studio installed.

Parameter Name	Description	Accepted Value(s)	Notes
NewReport	Launches the **Create New Report** dialog box, where you can begin the process of creating a new report.	True	This parameter also accepts False as a value. Note, however, that since the only reason for using this parameter is to create a new report, you should *never* need to set this parameter to False. Note also that to create a new report, you must have the full version of the SQL Server Business Intelligence Development Studio installed on your computer.
QueryKey	Creates a new report using the query specified by the key.	The key value of the query that you want to use for creating the new report.	To use this parameter for creating a new report, you must use NewReport=True. To create a new report, you must have the full version of the SQL Server Business Intelligence Development Studio installed.
QueryPath	Creates a new report using the query specified by the path.	The Catalog path and name of the query that you want to use for creating the new report.	To use this parameter for creating a new report, you must use NewReport=True. To create a new report, you must have the full version of the SQL Server Business Intelligence Development Studio installed.
ReportKey	Opens the Report Viewer, displaying the report specified by the ReportKey parameter.	The key value of the report that you want to open.	If the report is based on a query that contains a prompt, you can [pass values in for the prompts](#) via the URL.

Parameter Name	Description	Accepted Value(s)	Notes
ReportPath	Opens the Report Viewer, displaying the report specified by the ReportPath parameter.	The Catalog path and name of the report that you want to open.	If the report is based on a query that contains a prompt, you can pass values in for the prompts via the URL. If the report contains report prompts, you can pass values in for the prompts via the URL.

Examples of Report URLs

- meridium://Registry/ReportViewer?**NewReport=True**

  Opens the **Create New Report** dialog box, where you can choose to create a new report based on a query, a dataset, or an external template.

- meridium://Registry/ReportViewer?**ReportPath-h=Public\Meridium\Reports\SavedReport**

  Opens the report SavedReport in the Meridium APM Report Viewer.

- meridium://Registry/ReportViewer?**QueryPath-h=Public\Meridium\Queries\PumpQuery**

  Creates a new report using the query PumpQuery, which is stored in the Catalog folder \\Public\Meridium\Queries.

- meridium://Registry/ReportViewer?**DatasetKey=1234567**

  Creates a new report using the dataset that is identified by the key value 1234567.

- meridium://Registry/ReportViewer?**ReportWe-bService=http://CrystalReportsServer\[path to report]**

  Opens in the Report Viewer the report stored in the location *[path to report]* which resides on the CrystalReportsServer Application Server.

Parameters for the Strategy Rules URL

The URL for managing strategies, **meridium://StrategyManager/Strategies**, accepts the parameters described in the following table. Note that the URL path alone is not a valid URL. Use the information in the following table to choose parameters that will allow you to construct a usable link. For example of how you can combine these parameters to construct URLs, click here.

Parameter Name	Description	Accepted Value(s)	Notes
Action	Indicates what action to perform.	Create Execute Edit	• **Create:** Displays the **Create Strategy** page, where you can create a Strategy record. When you use Action=Create, you can optionally include the FamilyKey parameter to specify the family that you want to associate with the new Strategy record. If you *exclude* the FamilyKey parameter, a link constructed from a URL containing Action=Create will cause the **Entity Family** dialog box to appear, where you can select the family that you want to associate which the Strategy record. • **Execute:** Lets you execute a strategy rule. If you use Action=Execute, you must also include *either* the StrategyID parameter (to execute a single strategy rule) or the FamilyKey parameter (to execute all strategy rules associated with a specific family). You can optionally include the EntityKey parameter if you want to execute a specific strategy rule against a specific record. • **Edit:** Displays the **Edit Strategy** page, where you can modify an existing Strategy record identified by the StrategyKey parameter. If you use Action=Edit, you must also use View=EditCreate.

Parameter Name	Description	Accepted Value(s)	Notes
EntyKey	Identifies the record against which to execute the strategy rule.	The Entity Key of the desired record.	This parameter can be used when you execute strategy rules that are designed to be executed against a specific record in a family.
FamilyKey	Identifies the family that is associated with the selected Strategy record.	The Family Key of the family that is associated with the desired Strategy record.	This parameter is optional when you use Action=Create and allows you pass in the Family Key of the family that you want to associate with a new Strategy record, and the link will then bypass the **Entity Family** dialog box. This parameter can also be used with Action=Execute to execute ALL the strategy rules that are associated with a specific family.
StrategyKey	Identifies the Strategy record.	The Entity Key of the desired Strategy record.	This parameter is required when you use View=AuditInfo. This parameter is also required when both View=EditCreate and Action=Edit.
StrategyID	Identifies the Strategy record.	The Record ID of the desired Strategy record.	This parameter is required when Action=Execute.
View	Specifies what to display.	List AuditInfo EditCreate	• **List:** Displays the list of strategies. • **AuditInfo:** Displays the audit information for the Strategy record identified by the StrategyKey parameter. • **EditCreate:** When used with Action=Edit, displays the **Edit Strategy** page, which displays the Strategy record identified by theStrategyKey parameter.

Examples of the Strategy Rules URLs

- meridium://StrategyManager/Strategies?**View=List**

  Opens the **Strategy List** page.

- meridium://StrategyManager/Strategies?**View=AuditInfo&&StrategyKey=1234567**

  Displays the audit information for the Strategy cord identified by the Entity Key 1234567.

- meridium://StrategyManager/Strategies?**View=EditCreate**&StrategyKey-y=64251640458&**Action=Edit**

  Displays the **Edit Strategy** page, which displays the Strategy record with the Entity Key 64251640458.

- meridium://StrategyManager/Strategies?**Action=Create**

  Displays the **Entity Family** dialog box, where you can select a family that you want to associated with a new Strategy record.

- meridium://StrategyManager/Strategies?Action=Create&**FamilyKey=1234567**

  Displays the **Create Strategy** form, where you can create a new Strategy record that is associated with the family with the Family Key 1234567.

- meridium://StrategyManager/Strategies?**Action=Execute&FamilyKey=150517**

  Executes all the strategy rules that are associated with the family with the Family Key 150517.

- meridium://StrategyManager/Strategies?Action=Execute&**StrategyKey-y=2861953&EntyKey=598747**

  Executes the strategy rule that is associated with the Strategy record with the Entity Key 2861953 for the record with the Entity Key 598747.

Parameters for the Task Builder URL

The URL for the **Task Builder**, **meridium://TaskBuilder**, accepts the parameters described in the following table. Each parameter is optional. A link created from the path alone will launch the **Task Builder**, which will walk you through the process of creating a new Task record.

Parameter Name	Description	Accepted Value(s)
DatasheetID	Specifies the datasheet that you want to use for entering task data.	The datasheet ID of the desired datasheet.
EquipmentEntityKey	Specifies the Equipment or Functional Location record to which the Task record should be linked.	The Entity Key of the Equipment or Functional Location record.
FieldIDs	Specifies the fields that you want to populate automatically with customer-defined values.	The desired values.
NewTask	Specifies whether you are creating a new Task record or editing an existing Task record.	True (for a new Task record) False (for an existing Task record)
TaskFamilyKey	Specifies the family to which the Task record will belong.	The Family Key of the desired family.

When you use a URL to launch the **Task Builder**, your position in the builder will depend on which parameters were specified in the URL. For example, if the Entity Key is provided in the URL, the **Task Builder** will bypass the step where you have to select an Asset record. Instead, the Asset record with the provided Entity Key will automatically be selected in the **Task Builder**.

Examples of the Task Builder URL

- meridium://TaskBuilder

  Launches the **Task Builder** and displays the **Welcome** screen.

- meridium://TaskBuilder**?EquipmentEntityKey=2105162**

  Launches the **Task Builder** and displays the **Task Family Selection**screen, where you can select the desired Task family. The Equipment or Functional Location record with the specified Entity Key has already been selected.

- meridium://TaskBuilder?EquipmentEntityKey=2102254**&TaskFamilyKey=2021284**

  Launches the **Task Builder** and displays the **New or Existing Task Selection**screen, where you can choose to create a new Task record or modify an existing Task record. The list of existing Task records displays the Task records that have already been created for the specified Entity Key and Task family.

- meridium://TaskBuilder?EquipmentEntityKey=2102254&TaskFamilyKey=2021284 **&NewTask=true**& **DatasheetId=Calibration Task**

  Launches the **Task Builder**, which display a blank Calibration Task datasheet on the**Task Datasheet** screen.

- meridium://TaskBuilder?EquipmentEntityKey=2102254&TaskFamilyKey=2021284&NewTask=true& DatasheetId=Calibration Task**&MI_TASK_ID_C=12345&MI_TASK_DESC_ C=Task12345**

  Launches the **Task Builder** and displays the **Task Datasheet** screen. The Calibration Task datasheet is selected and is populated with the specified values.

Using the URL Builder to Create a Link to the Task Builder

When you create a URL to launch the **Task Builder**, you can specify parameters that determine what information will be automatically supplied to the **Task Builder**. In this way, you can easily create a task that is specific to a given Equipment record and task family. You can also determine which datasheet will be used by default and which fields, if any, will be pre-populated.

These instructions assume that you have already accessed the **URL Builder** and have chosen to create an internal Meridium APM URL.

To complete the steps in the URL Builder to create a link to the Task Builder:

1. In the **URL Builder**, on the **Select URL** screen, select **Task Builder** from the list, and click the **Next** button.

   The **URL Builder for Task Builder** screen appears, displaying options that let you pass in specific parameters to the **Task Builder**.

   All the parameters on this screen are optional. Without parameters, the **Task Builder** URL will launch the **Welcome** screen of the **Task Builder**, which will take you step-by-step through the process of creating a task. By passing in parameters, the **Task Builder** will skip any step for which information has already been supplied.

   > Note: At this point, you can click the **Finish** button in the **URL Builder** or complete as many of the following steps as desired.

2. Click the **Select Equipment** button.

   The **Find Items** window appears.

3. Perform a search to locate the desired piece of equipment.

   > ŒHint: The **Search In** field displays a list of families that participate in the Has Task relationship.

4. In the results list, select the desired record, and click the **Open** button.

   The equipment field in the **URL Builder** is populated with the record that you selected.

5. On the **URL Builder for Task Builder** screen, click the **Select Task Family** button.

   The **Select Family** dialog box appears.

6. In the list, and select the desired family.

7. Click **OK**.

The task family field in the **URL Builder** is populated with the family that you selected.

8. On the **URL Builder for Task Builder** screen, if you want the **Task Builder** to default to creating a new task, select the **Create a New Task** check box.

9. Click the **Select Datasheet** button.

> **Note:** This button is enabled only after you have selected a task family.

The **Select Family** dialog box appears.

10. In the list, select the datasheet that you want to use in the **Task Builder**, and click **OK**.

The datasheet field in the **URL Builder** is populated with the datasheet name that you selected.

11. Click the **Pre-Populate Data Fields** button.

The **Pre-Populate DataField Values** dialog box appears.

12. Specify the values that you want to use to pre-populate the datasheet in the **Task Builder**, and click **OK**.

13. On the **URL Builder for Task Builder** screen, click the **Finish** button.

Your URL is saved.

Parameters for the Administration Page URL

The URL for accessing the **Task Management Administration** page, **meridi-um://StrategyManager/Preferences**, requires the parameter described in the following table. [Click here for an example.](#)

Parameter Name	Description	Accepted Value(s)	Notes
View	Displays the **Task Management Administration** page.	Global	This parameter is required.

Examples of the Administration Page URL

- meridium://StrategyManager/Preferences**?View=Global**

  Displays the **Task Management Administration** page.

Parameters for the Task Record URL

The URL for accessing the **Task List** page, **meridium://StrategyManager/ProposedTasks**, accepts the parameters described in the following table. Note that none of these parameters are required; you can construct a functional link from the root URL. Note, however, that the function of the link will vary, depending on certain conditions. Therefore, we recommend that you specify the parameters that are necessary to invoke the action. For example of how you can combine these parameters to construct URLs, click here.

Parameter Name	Description	Accepted Value (s)	Notes
View	Specifies the page that you want to view.	List Executions Revisions	• **List:** Displays the **Task List** page • **Executions:** Displays the **Task Execution History** page for the Equipment or Functional Location record identified by the EntyKey parameter. Note that the EntyKey parameter is required if View=Executions. • **Revisions:** Displays the **Task Revision History** page for the Asset record identified by the EntyKey parameter. Note that the EntyKey parameter is required if View=Revisions.
Header_ Title	Specifies the text of the header that appears at the top of the workspace on the **Task List** page.	An alpha-numeric character string.	This parameter can be used only if View=List.

Parameter Name	Description	Accepted Value (s)	Notes
Query	Specifies the query that you want Meridium APM to run on the **Task List** page.	The Catalog path and name of the desired query.	This parameter can be used only if View=List.
EntyKey	Identifies the Entity Key of the Equipment or Functional Location record that you want to view as the root record on the **Task Execution History** page or the **Task Revision History** page.	The Entity Key of the desired Asset record.	This parameter is required if View=Executions or View-w=Revisions.
Action	Indicates the action that you want to perform.	AddNewTask	Displays the **Task Builder**, where you can create a new Task record.

Examples of Task Record URLs

- meridium://StrategyManager/ProposedTasks?**View=List**

  Displays the **Task List** page.

- meridium://StrategyManager/ProposedTasks?**View-w=Executions&EntyKey=1234567**

  Displays the **Task Execution History** page, where the root level of the tree displays the Equipment or Functional Location record with the Entity Key 1234567.

- meridium://StrategyManager/ProposedTasks?**View=Revisions**&EntyKey=1234567

  Displays the **Task Revision History** page, where the root level of the tree displays the Equipment or Functional Location record with the Entity Key 1234567.

- meridium://StrategyManager/ProposedTasks?View=List&**Header_Title=My Title**

  Displays the **Task List** page, where the header in the workspace contains the following text: My Title.

- meridium://StrategyManager/ProposedTasks?View=List&**Query=Public\All Inspection Tasks**

  Displays the results of the All Inspection Tasks query on the **Task List** page.

- meridium://StrategyManager/ProposedTasks?**Action=AddNewTask**

  Displays the **Task Builder**, where you can create a new Task record.

Parameters for the Task Types Record URL

The URL for accessing the **Task Types** page, **meridium://StrategyManager/TaskTypes**, accepts the parameters described in the following table. For example of how you can combine these parameters to construct URLs, [click here](#).

Parameter Name	Description	Accepted Value (s)	Notes
EntityKey	Identifies the Task Types record that you want to display.	The Entity Key of the desired Task Types record.	This parameter is required when View-w=Datasheet.
View	Specifies the format in which you want to view Task Types records.	List Datasheet	• **List:** Displays the list of Task Type records on the **Task Types** page. • **Datasheet:** Displays the **Edit Task Type** workspace of the **Task Types** page, where you can modify the Task Types record identified by the EntityKey parameter.

Examples of the Task Types Record URL

- meridium://StrategyManager/TaskTypes**?View=List**

  Displays the **Find Items** window, where you can search the Task Types family for a specific Task Types record.

- meridium://StrategyManager/TaskTypes?View=**Datasheet&EntityKey=1234567**

  Displays the **Edit Task Type** form, displaying the datasheet for the Task Types record identified by the Entity Key 1234567.

Parameters for the Task History URL

> **Note:** Beginning with V3.3.1, the Task History feature is obsolete. The Task History URL will continue to work for customers who were using the Task History feature in a previous release. We recommend, however, that you not use this URL if you are using Task Management for the first time beginning with V3.3.1.

The URL for accessing the **Task History** page, **meridium://StrategyManager/TaskHistory**, accepts the parameters described in the following table. For example of how you can combine these parameters to construct URLs, [click here](#).

Parameter Name	Description	Accepted Value(s)	Notes
View	Specifies that you want to display the **Task History** page.	List	This parameter is required.
EntyKey	Identifies the Equipment or Functional Location record whose linked Task History records you want to display on the **Task History** page.	A numeric Entity Key.	This parameter is required if you use the TskFamKey and Source parameters. Otherwise, this parameter is optional.
Source	Specifies the location from which the **Task History** page is being accessed.	RecordManager	This parameter is optional.
TskFamKey	Identifies the family of the current Task record. You would use this parameter if you were viewing a Task record in Record Manager and wanted to access the **Task History** page directly from the Record Manager (i.e., via a link on the **Associated Pages** menu).	A numeric Family Key.	This parameter is optional. If used, it must be used in conjunction with the Source parameter.

Examples of the Task History URL

- meridium://StrategyManager/TaskHistory?**View=List**

  Displays the **Find Items** window, which lets you search for an Asset record and display the linked Task History records on the **Task History** page.

- meridium://StrategyManager/TaskHistory?View=List&**EntyKey=2004743**

  Opens the **Task History** page, displaying the Task History records that are linked to the record identified by the Entity Key 2004743.

- meridium://StrategyManager/TaskHistory?View=List&EntyKey=2004743&
  TskFamKey=2847737&**Source=RecordManager**.

  From Record Manager, opens the **Task History** page, which displays Task History records that are linked to the record identified by the Entity Key 2004743. This URL also specifies that the current record in the Record Manager belongs to the Task family identified by the Family Key 2847737.

Parameters for the Timeline URL

The URL for accessing the **Timeline** page, **meridium://StrategyManager/Timeline**, accepts the parameters described in the following table. For example of how you can combine these parameters to construct URLs, click here.

Parameter Name	Description	Accepted Value (s)	Notes
EntityKey	Identifies the Equipment or Functional Location record whose linked Task records you want to display the Timeline.	A numeric Entity Key.	This parameter is required whenever QueryType=Equipment.
FamilyKey	Identifies the family of the Equipment or Functional Location record specified by the EntityKey parameter.	An alpha-numeric Family Key.	This parameter is required whenever QueryType=Equipment.
QueryPath	Specifies which query you want to run for the purpose of returning records on the Timeline.	The Catalog path and name of the desired query.	This parameter is required whenever QueryType=UserQuery.

Parameter Name	Description	Accepted Value (s)	Notes
QueryType	Specifies the way in which you want to locate records to display on the Timeline.	TaskFamily Strategy Equipment UserQuery	• **TaskFamily:** Displays Task records in the family identified by the TaskFamilyID parameter. • **Strategy:** Displays Task records that were created as a result of the Strategy record identified by the StrategyKey parameter. • **Equipment:** Displays Task records that are linked to the Equipment or Functional Location record identified by the EntityKey and FamilyKey parameters. • **UserQuery:** Displays Task records that are returned by the query identified by the QueryPath parameter.
StrategyKey	Specifies the Strategy record that generated the Task records that you want to display on the Timeline.	A numeric Entity Key.	This parameter is required whenever QueryType=Strategy.
TaskFamilyID	Specifies the Task family whose Task records you want to display on the Timeline.	An alpha-numeric family ID.	This parameter is required whenever QueryType=TaskFamily.
View	Specifies to display the **Timeline** page.	Chart	This parameter is required if View w=QueryType.

Examples of the Timeline URL

- meridium://StrategyManager/TimeLine

  Opens the **Task Query Builder**, which lets you choose options for accessing the Timeline.

- meridium://StrategyManager/Timeline**?View-w=Chart&QueryType=TaskFamily&TaskFamilyID=MI_TASKCALB**

  Displays on the Timeline Task records in the Task Family with the ID MI_TASKCALB.

- meridium://StrategyManager/TimeLine?View=Chart&QueryType=**Strategy&StrategyKey=2813931**

  Displays on the Timeline Task records generated from the Strategy record identified by the Entity Key 2813931.

- meridium://StrategyManager/TimeLine?View=Chart&QueryType=**User-Query&QueryPaty=Public\Meridium\Queries\TimelineQuery**

  Displays on the Timeline Task records that are returned by the Timeline Query, which is stored in the Catalog folder \\Public\Meridium\Queries.

Dashboard

A Meridium APM Catalog item that can be used to:

- Display summary information about the data in your Meridium APM system.

    -and-

- Navigate to other areas in the Meridium APM Framework.

Dataset

A fixed set of information, stored as a Catalog item, that can be viewed and modified in a grid, where columns and rows form intersecting cells that store data.

Graph

A visual representation of data that is based on a query or dataset.

Query

A tool that you can use to construct a question or request of the database and then view the answer that is returned in the form of query results.

Report

A formatted view of data that exists in the Meridium APM database. Reports provide a way of formatting information for display, export, and printing purposes.

Strategy Rule

Rules that you create using VSTA and that define actions that you want the Meridium APM system to perform.

Task Management

A tool that allows you to manage all Task, Task Revision, and Task Execution records that exist in the database.