Meridium APM Configuration Tools

V3.6.1.7.0

meridium

Meridium APM Configuration Tools

V3.6.1.7.0

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/secure/documentation/WebHelp/Home.htm).

> 🗎 **Note**: If you do not have access to the Meridium APM Documentation Website, contact GE Global Support (https://www.ge.com/digital/asset-performance-management).

Table of Contents

Creating New Families

The following instructions provide details on how you can add a new entity or relationship family to the family hierarchy. Note that adding a family to the hierarchy is just the first of several steps that may be involved in creating a family that will be accessible to Meridium APM users and can be used for storing records, including:

- Defining fields for the family.
- Defining rules for family fields and rules for the family.
- Creating the physical tablespace for the family.
- Compiling the family rule code.
- Creating a datasheet for the family (entity families only).
- Granting users permission to the family.
- Configure relationship definitions involving the family. This step is not required for creating a fully functional family, but it is a step that you will likely want to perform for most of the entity families that you create.

> 🔲 **Note:** After you create a new family, it may be necessary to recycle the cache on the Application Server before you can create a new report on that family in the Meridium APM Framework application. For details on recycling the cache, see the Meridium APM Installation, Upgrade, and System Administration Help.

To add a new family to the hierarchy:

1. In the Configuration Manager, in the **Entity Family/Relationship Family** pane, on the **Entity Family** tab, click the family under which you want to add the new family. The new family will be added as a subfamily under the family that you select. To add the new family at the root level, select the **Root** folder.

2. Click the **Add** button, which appears at the bottom of the **Entity Family/Relationship Family** pane.

   The **Add Entity Family** dialog box appears.

3. In the **Caption** text box, type the name of the family. This label will appear on the interface to identify the family. The caption is required and must be unique within the system.

   The name that you type in the **Caption** text box automatically appears in the **ID** text box.

4. Modify the value in the **ID** text box, if desired. The ID is required and must be unique as compared to other family IDs in the database, but it can be the same as the Caption.

   > 🔲 **Note:** The family ID cannot start with a number.

5. Click the **Save** button.

A confirmation message appears, indicating that the family has been saved.

6. Click **OK**.

The new family appears in the hierarchy on the **Entity Family** tab, sorted alphabetically.

At this point, you can configure the settings available in the **Family Information**, **Physical Storage**, and **Tasks** sections that appear on the right side of the **Meridium APM Configuration Manager** main window.

Displaying the Audit Information for a Family

You can view the date of the last update to the family and the user who did so by choosing to display the audit information.

To display the audit information for a family:

1. In the Configuration Manager, in the **Entity Family/Relationship Family** pane, click the **Entity Family** tab.

2. Select the family whose audit information you want to view.

3. In the **Family Information**, section on the right side of the window, click the **Audit Information** link.

    The **Audit Information** dialog box appears, showing when the family was last modified and who modified it.

    Audit Information

    | Last Updated: | Friday, May 20, 2005 11:12:15 AM |
    | Last Updated By: | Administrator Meridium |
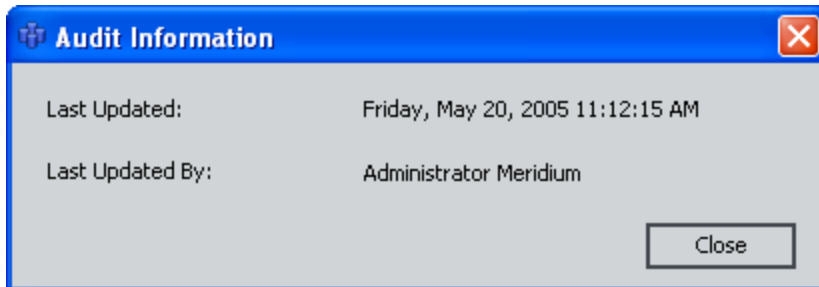
    Close

4. When you are finished viewing the audit information, click the **Close** button.

Deleting Families from the Database

If necessary, you can delete a family from the database. Note that when you delete a family, the following items will also be deleted:

- Subfamilies
- Successor/predecessor relationship definitions
- Family rules
- Family datasheets
- Family fields
- Physical tables and views associated with that family

> 🛈 **Note:** Only non-baseline families can be deleted. If you attempt to delete a baseline family, after asking for confirmation, an error message will appear, and the family will not be deleted. For baseline families that have been delivered with read-only properties, the **Delete Family** link will be disabled.

To delete a family from the database:

1. In the Configuration Manager, in the **Entity Family/Relationship Family** pane, click the **Entity Family** tab.

2. Select the family that you want to delete.

3. In the **Tasks** section on the right side of the window, click the **Delete Family** link.

   A message appears, asking if you really want to perform the deletion. The message provides a list of all the things that will be deleted when you delete the family.

4. Click the **Yes** button.

   The family is deleted from the database.

About Managing Physical Tables

Each family that you create in the Configuration Manager has a corresponding physical database table and database view. Creating a family in the Configuration Manager, however, does not automatically create the physical tablespace for that family. After you create a family, you will need to create the database table and database view manually using the **Manage Physical Tables** tool in the Configuration Manager. In addition, you will need to use this tool whenever you add a field to an existing family, delete a field from a family, or make changes that necessitate the re-creation of database views.

The **Manage Physical Tables** tool provides the **Script Builder**, which takes you step-by-step through the process of creating a script to modify the physical database. The **Script Builder** helps you identify which families require changes to the database, and then it generates a script to make the changes for the families of your choice. The **Script Builder** lets you:

- Generate a script to create or modify physical tablespace.
- Generate a script to delete physical tablespace.
- Generate a script to re-create database views.
- Run a saved script.

> 🛈 **Note:** Based on your selections, the **Script Builder** automatically generates a script that will modify the content of the Meridium APM database. After the script has been generated, the **Script Builder** displays the code so that you can view the script and, if necessary, modify it. In most cases, you should not modify the code that is generated by the **Script Builder**. Doing so could result in changes that will cause the Meridium APM system to generate errors or behave unexpectedly. Any changes that are needed to the code generated by the **Script Builder** should be made *only* by users who have a working knowledge of SQL.

After you create a script using the **Script Builder**, you will have the option of running the script immediately or saving the script for future use. Any script that you save can be run at a later time via the **Script Builder**.

> ⚠ IMPORTANT: Any changes that are made to the database when you run a script via the **Script Builder** are *permanent*.

Creating and Modifying Physical Tables

The following instructions provide details on generating a script to create or modify database tables and views for one or more families. You will need to run this type of script to make the associated database changes whenever you create a new family, add fields to an existing family, or delete fields from a family.

To create a script that will create or modify physical tables:

1. In the Configuration Manager, on the main menu, click **Tools**, and then click **Manage Physical Tables**.

   The **Script Builder** appears.
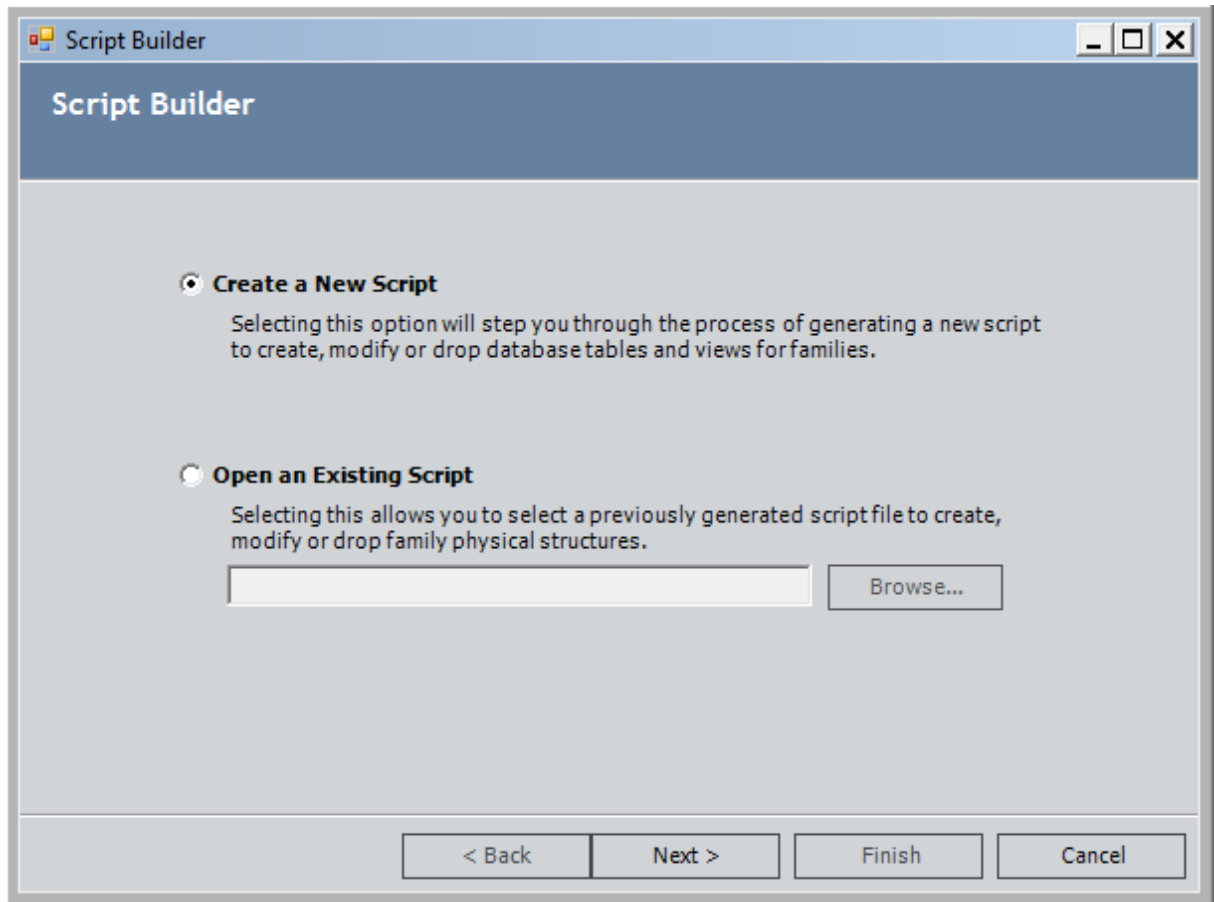
2. Accept the default selection, **Create a New Script**, and click the **Next** button.

   The **Script Builder** displays the **Select the Type of Script to Generate** screen.
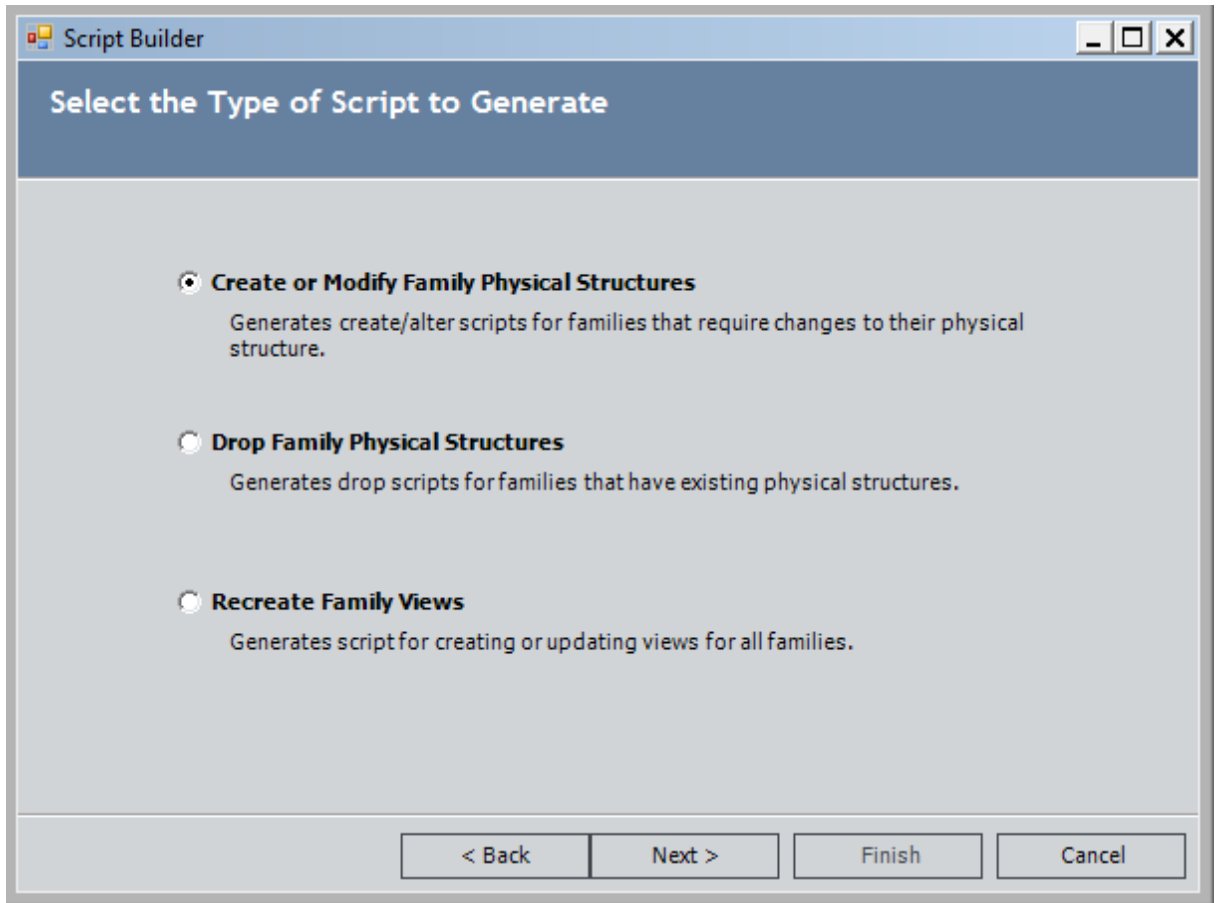
3. Accept the default selection, **Create or Modify Family Physical Structures**, and click the **Next** button.

   A progress bar appears as the system retrieves family information from the database. After the information has been retrieved, the **Script Builder** displays the **Select Items to Create/Alter** screen.
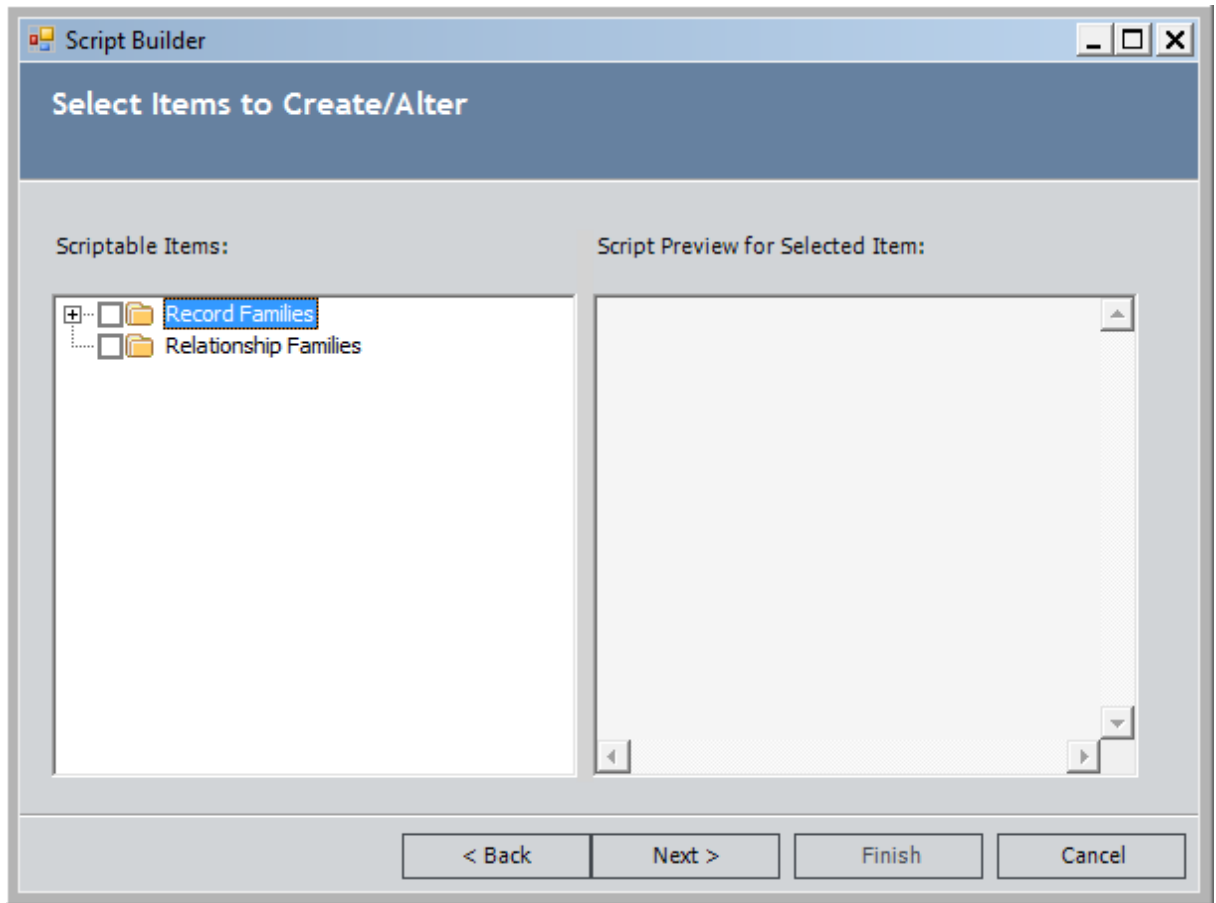
4. In the **Scriptable Items** area on the left side of the screen, expand the tree and select the check boxes for the families that you want to include in the script, as shown in the following image.
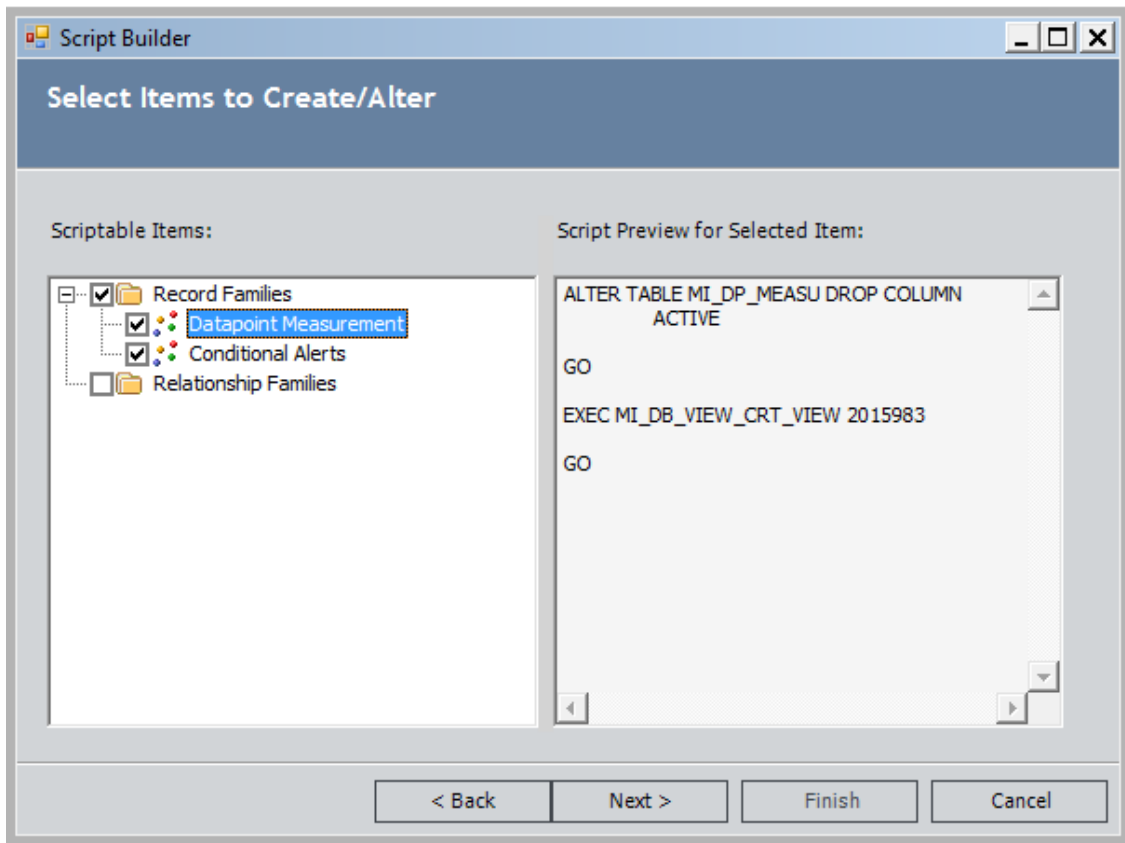
The list contains all the families whose physical tables and views need to be modified as a result of changes made via the Configuration Manager. You can highlight any family in the tree on the left to display in the area on the right the specific changes that will be made if you select that family to be included in the script.

> ⓘ **Note:** Selecting the check box for a given family will select *that* family and any family *above* that family. Subfamilies will not be selected automatically. You will need to expand the tree and manually select the check box for each subfamily whose physical tables you want to add or modify.

5. Click the **Next** button.

    The **Script Preview** screen appears, displaying the contents of the SQL code that was generated based on your selections.

> ⓘ **Note:** On this screen, you can make changes to the script, if necessary. We recommend, however, that you make change only if they are absolutely necessary and only if you have a working knowledge of SQL.

6. If desired, click the **Save this Script** link to save the script so that it can be executed later. When the **Save As** dialog box appears, navigate to the desired location, and save the script. After you save the script, you can click **Cancel** to exit the **Script Builder** or, if you want to run the script, you can proceed with the next step.

7. Click the **Next** button.

   A message appears, indicating that running the script will modify the physical database.

8. Click the **Yes** button to run the script. You can click the **No** button if you do not want to run the script.

   The script runs. When all of the database changes have been made, the **Script Execution Results** screen appears. The value in the **Result** column indicates the success or failure of the database change that the script was designed to make. If desired, you can click the **Save Execution Results to a File** link to save the results that are displayed on this screen.

9. Click the **Finish** button to close the **Script Builder**.

•

Deleting a Table

The following instructions provide details on generating a script to delete database tables and views for one or more families. Because drop scripts delete tables and views but do not delete the metadata (i.e., you can still access the family and fields in the Configuration Manager), a drop script is particularly useful in cases where you want to delete data from the database but not completely delete the family.

For example, you may want to delete tables and views if you change the name of the family table or view or if you make significant changes to the family fields. After you drop the existing tables, you can create new tables and views using the updated metadata. Dropping the tables and views and recreating them is the cleanest way to apply major changes to the database, provided that the existing tables contain no data.

Note that you do not need to run a drop script to delete a family. When you delete a family via the Configuration Manager, all the associated database tables and views are also deleted.

> ⚠ **IMPORTANT:** Deleting a family via a drop script deletes the physical database table and ALL records belonging to the family. These changes are permanent.

To create a new Drop script:

1. In the Configuration Manager, on the main menu, click **Tools**, and then click **Manage Physical Tables**.

    The **Script Builder** appears.
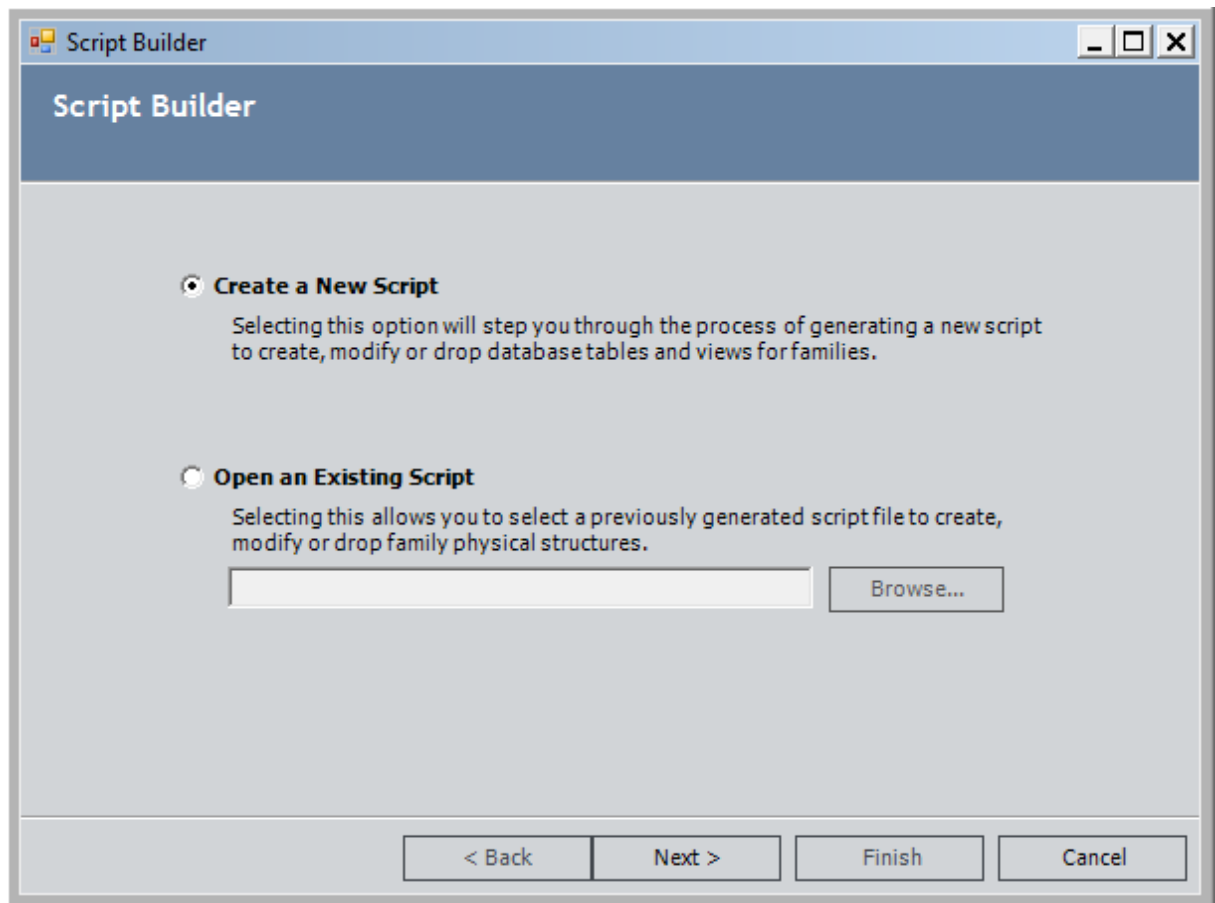
2. Accept the default selection, **Create a New Script**, and click the **Next** button.

   The **Script Builder** displays the **Select the Type of Script to Generate** screen.
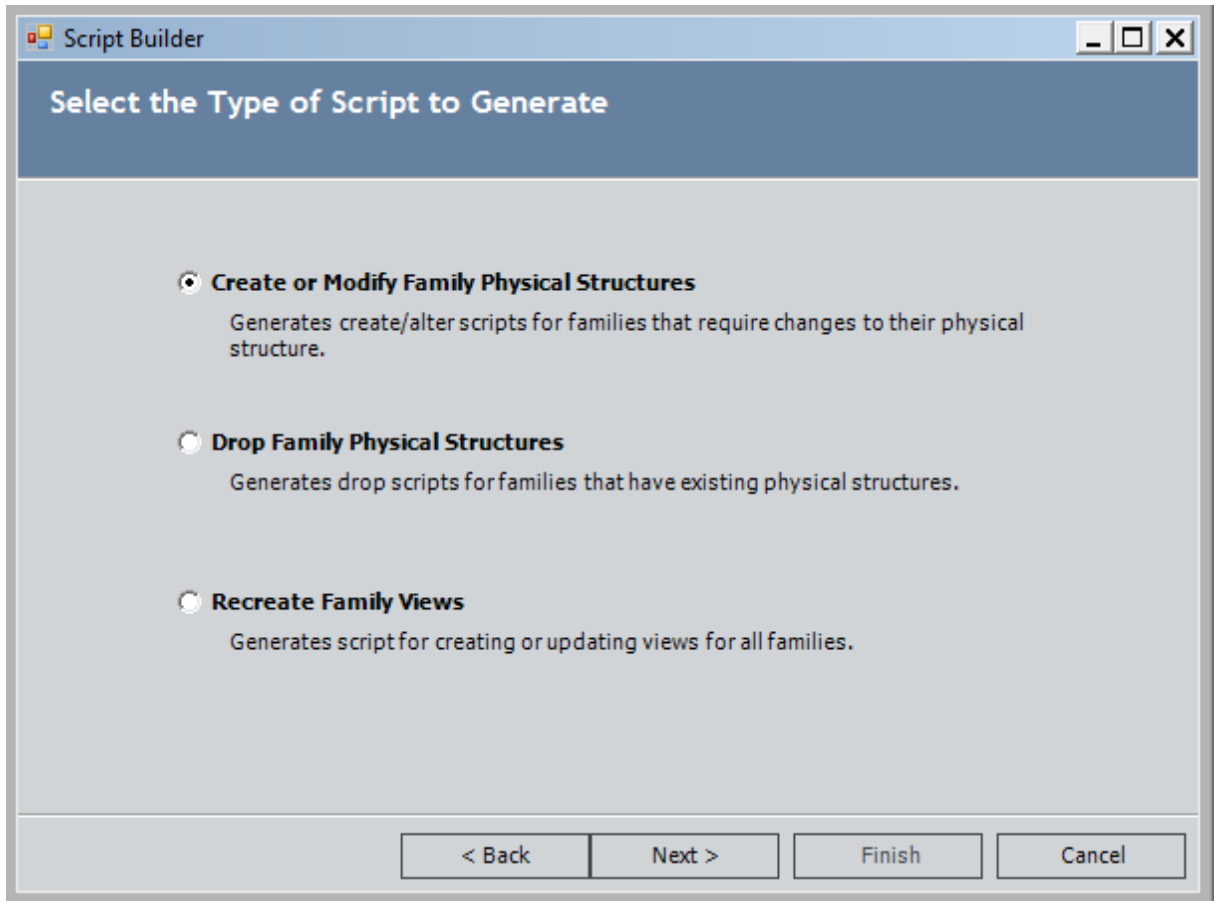
3. Select the **Drop Family Physical Structures** option, and click the **Next** button.

   A progress bar appears as the system retrieves existing family information from the database. The **Script Builder** then displays the **Select Items to Drop** screen.

4. In the **Scriptable Items** area on the left side of the screen, expand the tree and select the check boxes for the families that you want to include in the script. For example, in the following image, the Alert family has been selected.
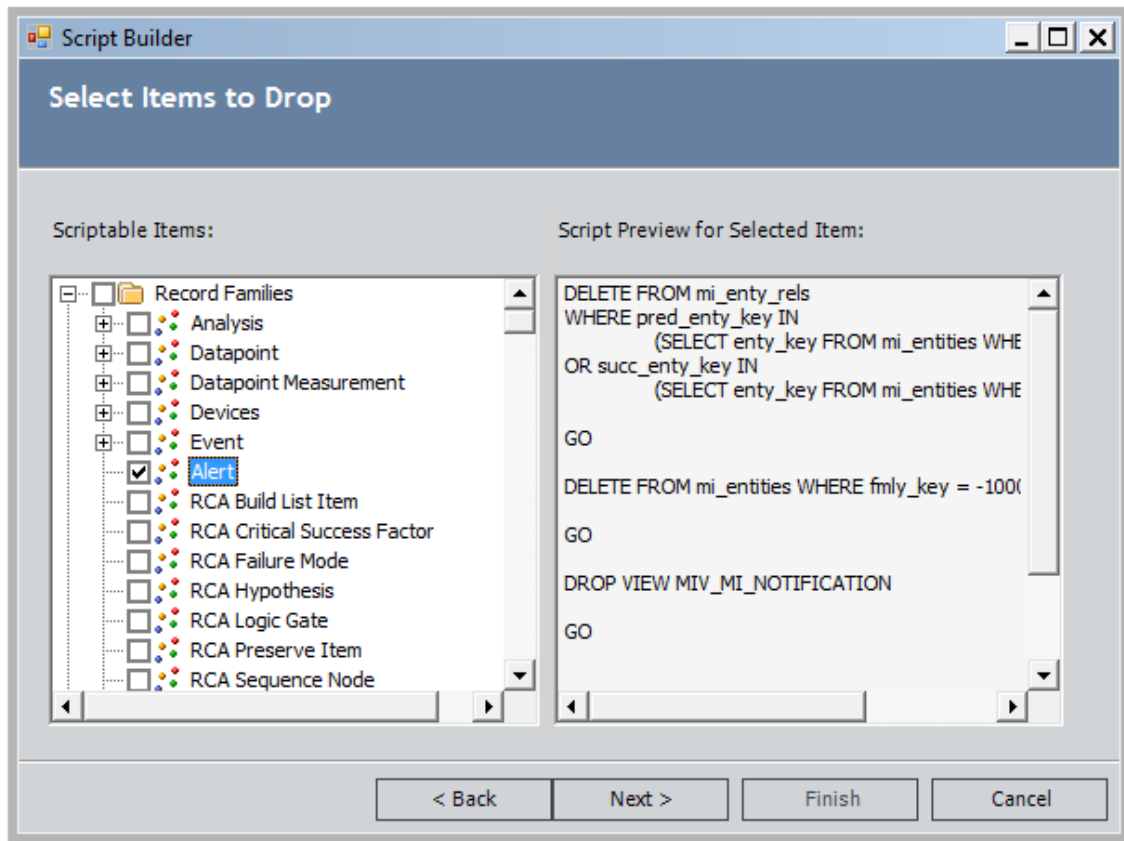
The list contains all the families that currently have physical tables or views in the database. You can highlight any family to display in the area on the right the specific changes that will be made if you select that family to be included in the script.

> ⓘ **Note:** Selecting the check box for a given family will select *that* family and any family *above* that family. Subfamilies will not be selected automatically. You will need to expand the tree and manually select the check box for each subfamily whose physical table you want to delete.

5. Click the **Next** button.

The **Script Preview** screen appears, displaying the contents of the SQL code that was generated based on your selections.

> **Note:** On this screen, you can make changes to the script, if necessary. We recommend, however, that you make change only if they are absolutely necessary and only if you have a working knowledge of SQL.

6. If desired, click the **Save this Script** link to save the script so that it can be executed later. When the **Save As** dialog box appears, navigate to the desired location, and save the script. After you save the script, you can click **Cancel** to exit the **Script Builder** or, if you want to run the script, you can proceed with the next step.

7. Click the **Next** button.

   A message appears, indicating that running the script will modify the physical database.

8. Click the **Yes** button to run the script. You can click the **No** button if you do not want to run the script.

   The script runs. When all of the database changes have been made, the **Script Execution Results** screen appears. The value in the **Result** column indicates the success or failure of database change that the script was designed to make. If desired, you can click the **Save Execution Results to a File** link to save the results that are displayed on this screen.

9. Click the **Finish** button to close the **Script Builder**.

Recreating Family Views

The **Script Builder** provides the option to recreate database views, if necessary. You will want to generate a script to recreate database views whenever you make changes that will affect database views. A database view represents all fields for a given family, including fields that are defined directly on the family and stored in the family's physical table and fields defined on families, which are stored in the families' physical tables, but are spread down to the subfamily. Therefore, you will want to recreate views, for example, when you modify the properties of spread fields. You should also recreate database views if you modify the database view name for a family.

To create a script to recreate database views:

1. In the Configuration Manager, on the main menu, click **Tools**, and then click **Manage Physical Tables**.

   The **Script Builder** appears.
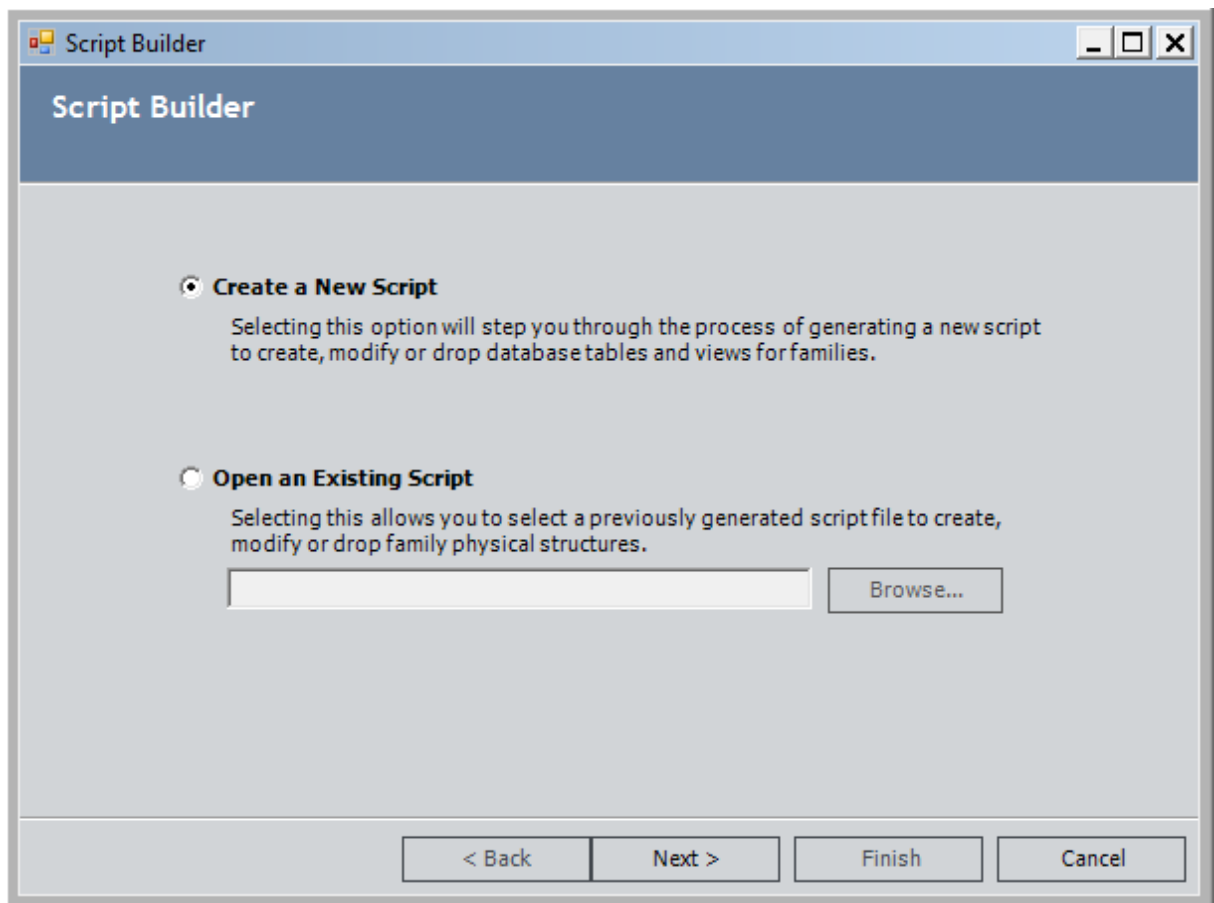
2. Accept the default selection, **Create a New Script**, and click the **Next** button.

   The **Script Builder** displays the **Select the Type of Script to Generate** screen.
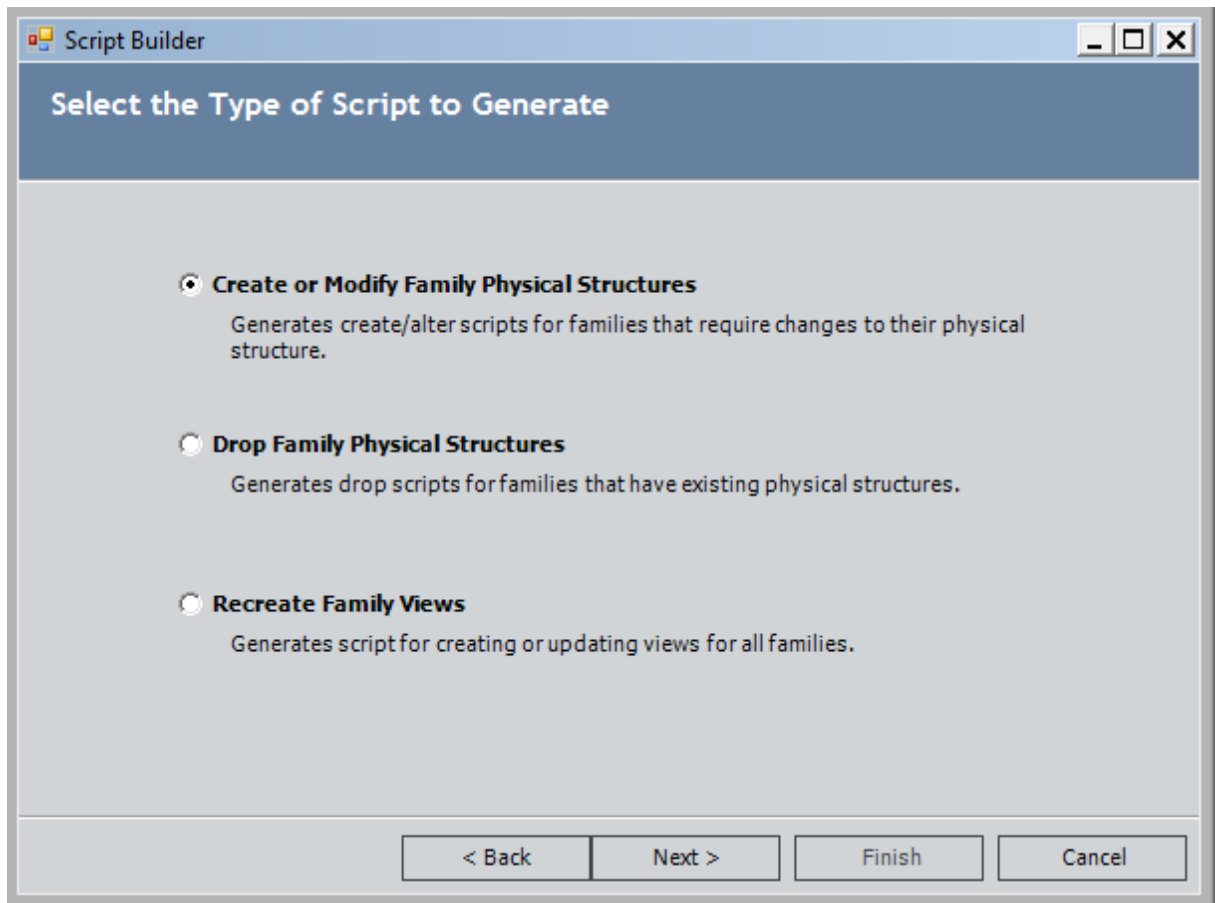
3. Select the **Recreate Family Views** option, and click the **Next** button.

   The **Script Preview** screen appears, displaying the SQL script that will be executed to recreate all views in the database.
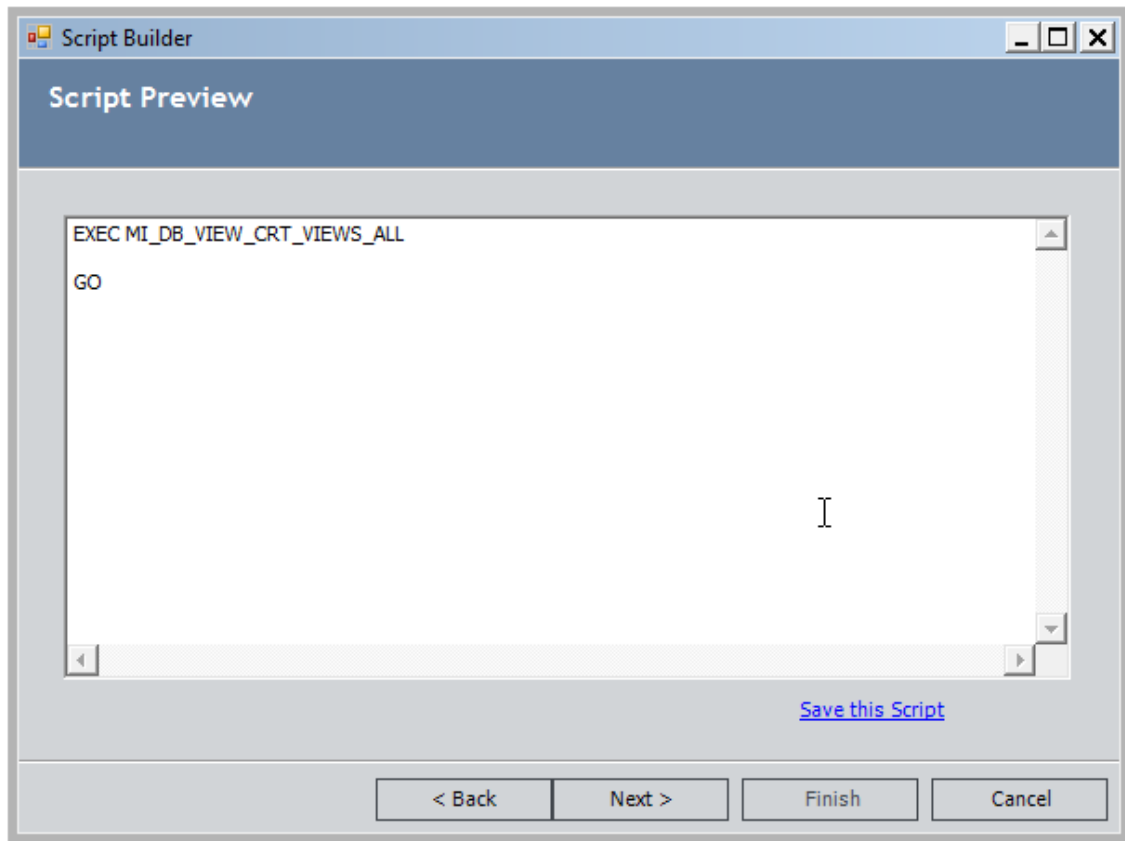
> 📄 **Note:** On this screen, you can make changes to the script, if necessary. We recommend, however, that you make change only if they are absolutely necessary and only if you have a working knowledge of SQL.

4. If desired, click the **Save this Script** link to save the script so that it can be executed later. When the **Save As** dialog box appears, navigate to the desired location, and save the script. After you save the script, you can click **Cancel** to exit the **Script Builder** or, if you want to run the script, you can proceed with the next step.

5. Click the **Next** button.

   A message appears, indicating that running the script will modify the physical database.

6. Click the **Yes** button to run the script. You can click the **No** button if you do not want to run the script.

   The script runs. When all of the database changes have been made, the **Script Execution Results** screen appears. If desired, you can click the **Save Execution Results to a File** link to save the results that are displayed on this screen.

7. Click the **Finish** button to close the **Script Builder**.

•

Running a Saved Script

Whenever you create a script using the Script Builder, you have the option of saving the script to a file. The Script Builder provides an option to open any saved script and run it. Additionally, if needed, you can make changes to the script and re-save it. The following instructions provide details on opening and running a saved script.

To open and run a saved script:

1. In the Configuration Manager, on the main menu, click **Tools**, and then click **Manage Physical Tables**.

   The **Script Builder** appears.
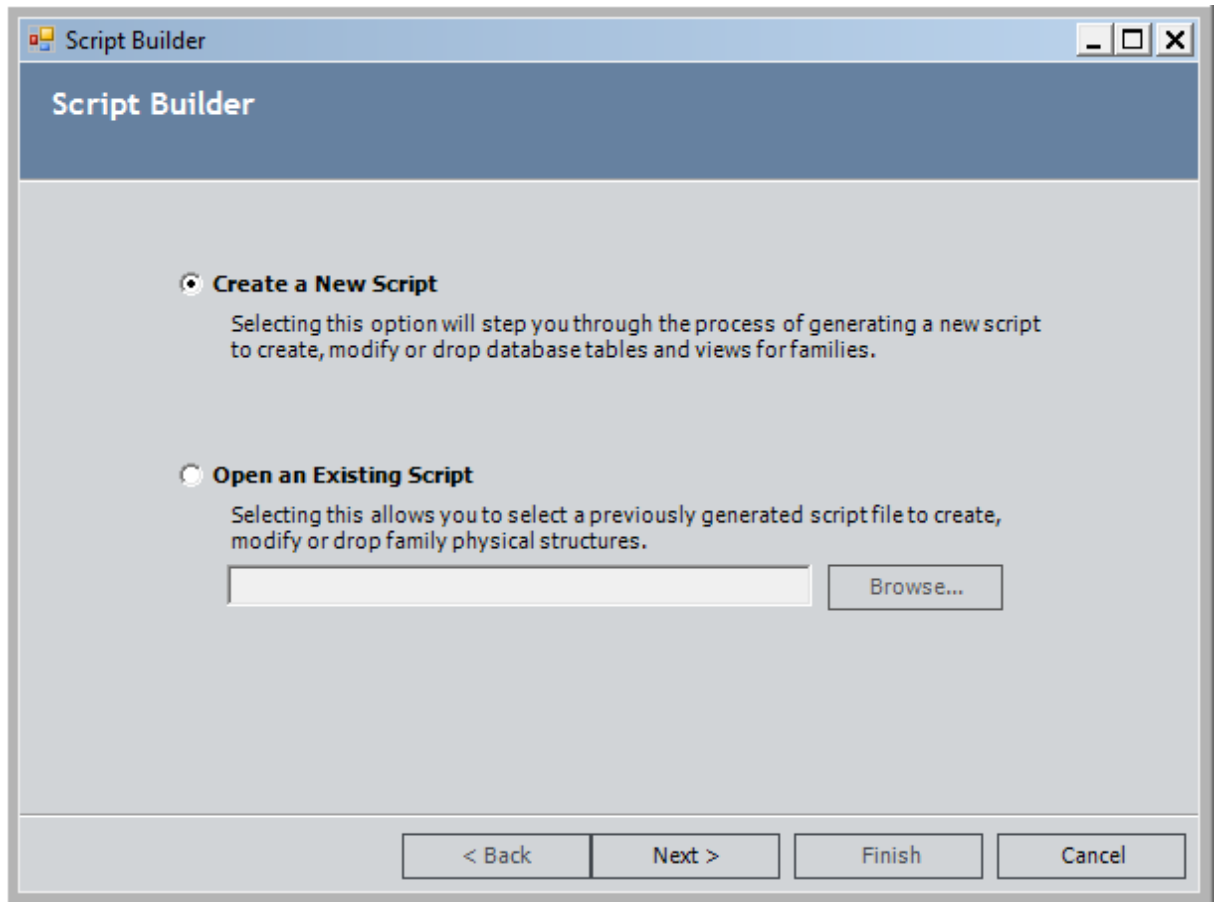
2. Select the **Open an Existing Script** option.

   The **Browse** button becomes enabled.
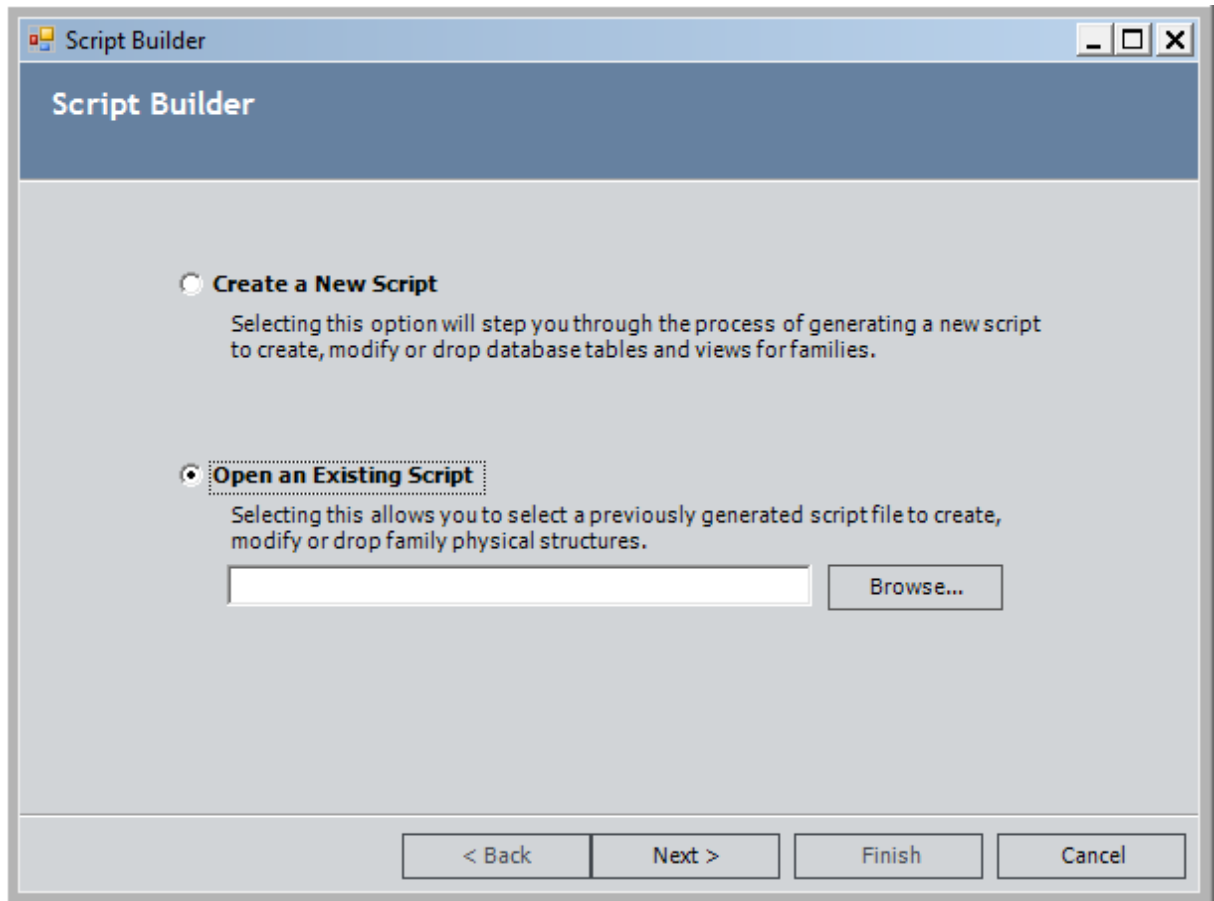
3. Click the **Browse** button.

   The **Open an Existing SQL Script** dialog box appears.

4. Navigate to and select the script that you want to open.

5. Click the **Open** button.

   The text box on the **Script Builder** window is populated with the script that you selected.

6. Click the **Next** button.

   The **Script Preview** screen appears, displaying the content of the saved script.

   > 🛈 **Note:** On this screen, you can make changes to the script, if necessary. We recommend, however, that you make change only if they are absolutely necessary and only if you have a working knowledge of SQL. You can click the **Save this Script** link to save any changes that you make to the script.

7. Click the **Next** button.

A message appears, indicating that running the script will modify the physical database.

8. Click the **Yes** button to run the script. You can click the **No** button if you do not want to run the script.

   The script runs. When all of the database changes have been made, the **Script Execution Results** screen appears. The value in the **Result** column indicates the success or failure of database change that the script was designed to make. If desired, you can click the **Save Execution Results to a File** link to save the results that are displayed on this screen.
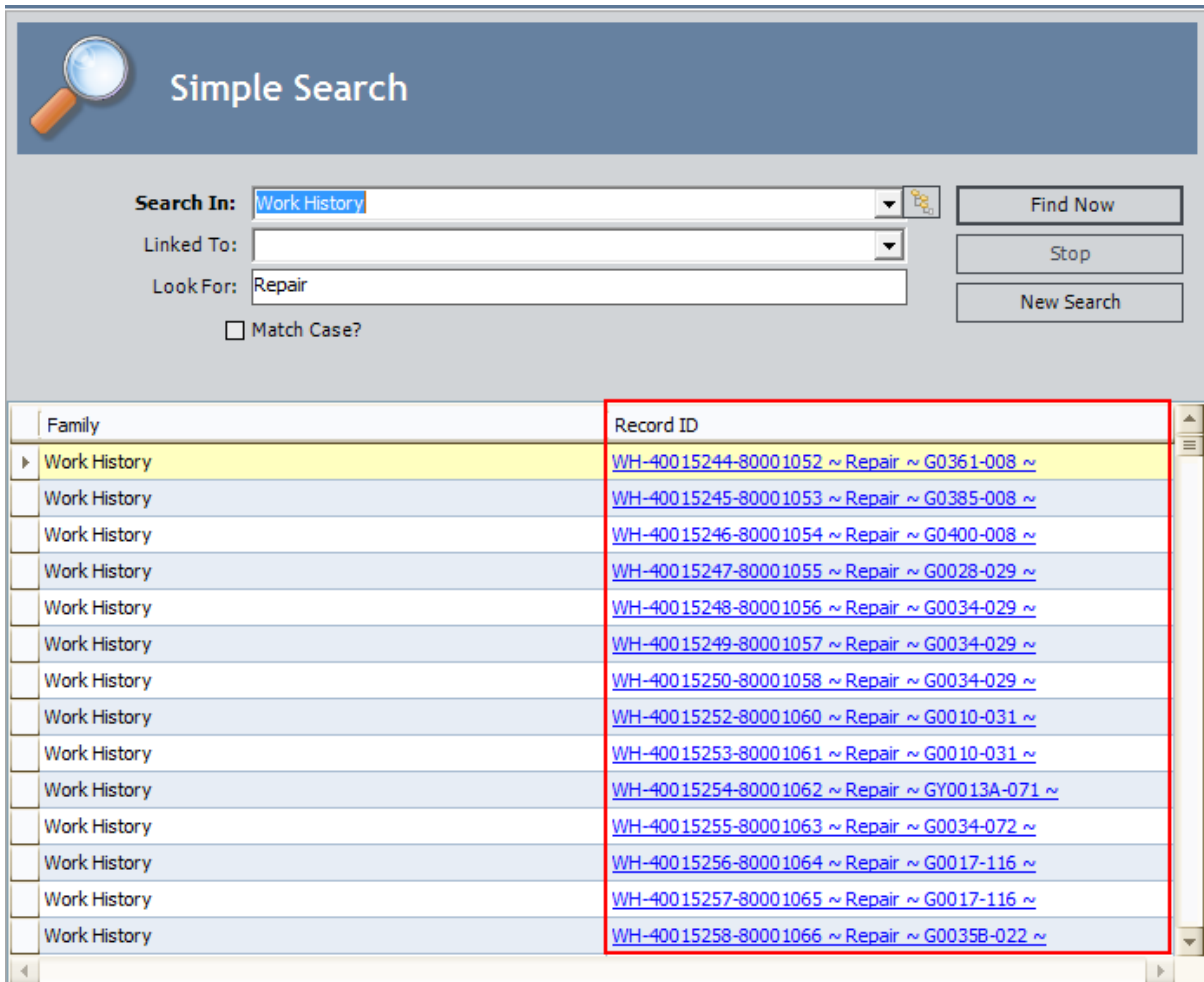
9. Click the **Finish** button to close the **Script Builder**.

About ID Templates

The Meridium APM system uses ID Templates to construct Record IDs and index records in the system. When a record is created in the system, a record ID is created using the values stored in the fields that are included in the ID Template. The Record ID is stored in the ENTY_ID system field of that record. After the Record ID has been created, it can be used to search for that record in the Meridium APM Framework application and in the Meridium APM Web Framework.

For example, when you perform a Simple Search in the Meridium APM Framework application and specify a value in the **Look For** text box, the Meridium APM system searches for the specified keyword(s) in the Record ID's of records in the database. Your search results will include records that contain your keyword(s) in their Record IDs.

In the following image, the value *Repair* appears in the **Look For** text box, so the results include only records with the value *Repair* in the Record ID.

Therefore, when constructing ID Templates, be sure to choose fields that will be useful for searching. Additionally, you will want to arrange the fields in a way that will display identifying information to users in a logical manner.

Viewing and Modifying the ID Template

The ID Template specifies how fields will be included and arranged to construct the Record ID of records belonging to a given family.

Note that if you create an ID Template that includes only one field, you can modify the ID Template only by adding another field to it. You cannot remove a single field from an ID Template.

> ⓘ **Note:** Record IDs that are constructed based upon the ID Template are stored in the ENTY_ID field in the database. The character limit of the ENTY_ID field is 255. Record IDs will be truncated, if necessary, to fit in the space allowed. For example, if the ID Template is made up of two character fields, each of which has a character limit of 255, the Record ID has the potential to contain 510 characters but will always be truncated so that it does not exceed 255 characters.

To view or modify the ID Template defined for a family:

1. In the Configuration Manager, in the **Entity Family/Relationship Family** pane, click the **Entity Family** tab.

2. Select the family whose ID Template you want to modify.

3. In the **Tasks** section on the right side of the window, click the **Manage ID Template** link.

   The **Manage ID Template** window appears.
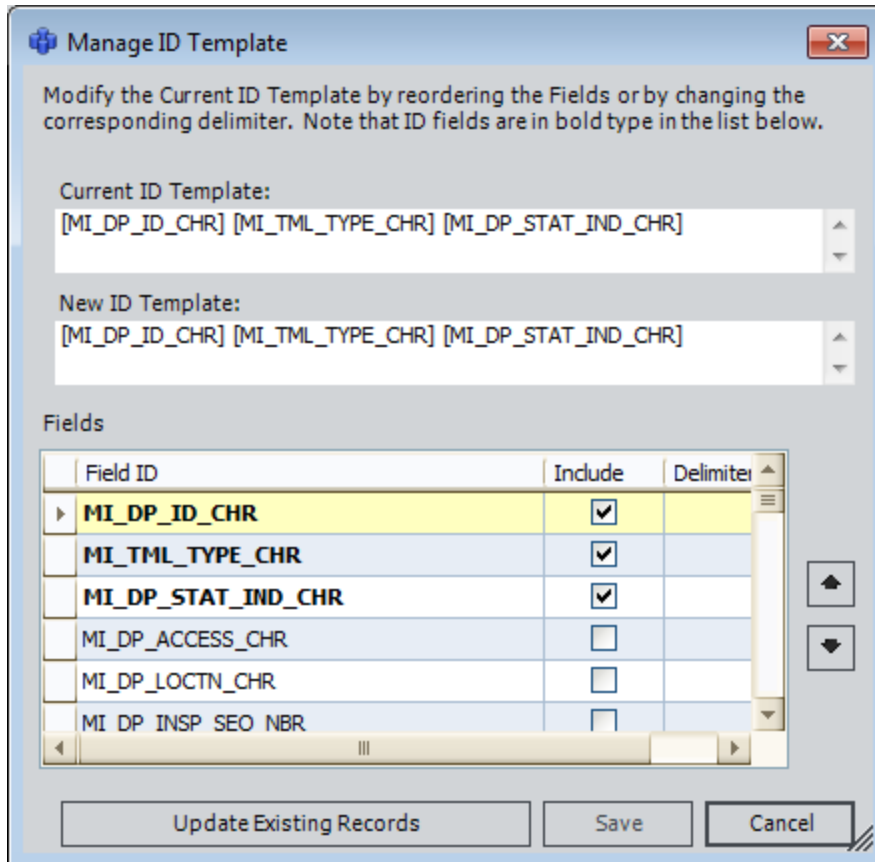
The **Current IDTemplate** area displays the current ID Template. The **New ID Template** area displays changes that you make to the ID Template. As the field IDs are rearranged, the order of fields within the **New ID Template** changes. A list of fields defined for the family is included along with the current ID Template; ID fields appear in bold text.

4. If desired, change the order of the fields that will be used to construct the Record ID by clicking the up and down arrow buttons.

5. Select the **Include** check box to include a given field in the ID Template.

> 🛈 **Note:** The ID Template itself cannot exceed 255 characters in length. This means that you cannot include fields whose Field IDs together exceed 255 characters.

6. Clear the **Include** check box for any currently included field that you want to *remove* from the ID Template.

7. If desired, in the **Delimiter** column, type a delimiter for each field. The delimiter will be used to separate different fields. For example, you might use an ellipsis or a dash.

8. Click the **Save** button.

The Meridium APM system saves your changes and displays a message indicating that the ID Template was updated successfully.

9. Click **OK**.

   A message appears, reminding you that before you can update existing records based upon the new ID Template, all the necessary columns must exist in the physical table. If you added new fields to family and then modified the ID Template without first <u>updating the physical tables</u>, the necessary columns may not exist.

10. Click **OK**.

    A message appears, asking if you want to update existing records based upon the new ID Template.

11. If you want to update existing records, click the **Yes** button.

    The Meridium APM system will update the records, and then the **Manage ID Template** window will close automatically.

    -or-

    If you do not want to update existing records, click the **No** button.

    The **Manage ID Template** window closes. If the necessary columns do not already exist in the physical table, you will need to click the **No** button, <u>update the physical tables</u>, and afterwards <u>return to the **Manage ID Template** window to update existing records</u>.

Updating Existing Record IDs

When you create a new record, the Record ID for that record is constructed using the ID Template that exists for the family at that time. If you make changes to a family's ID Template after records already exist in that family, you may want to update the Record IDs in existing records with new IDs that are based upon the updated ID Template.

When you make changes to a family's ID Template and save those changes, a prompt appears, asking you if you want to update existing records with new IDs based upon the new ID Template. If you choose not to update existing records at that time, you can return to the **Manage ID Template** window later to make the updates.

> ⓘ **Note:** Whether you update records immediately after making changes or at a later time, we recommend that you *always* update existing records whenever you modify the ID Template so that all the record IDs for a given family are in sync with one another.

To update existing records with new IDs based upon changes to the ID Template:

1. In the Configuration Manager, in the **Entity Family/Relationship Family** pane, click the **Entity Family** tab.

2. Select the family whose ID Template you want to modify.

3. In the **Tasks** section on the right side of the window, click the **Manage ID Template** link.

   The **Manage ID Template** window appears, displaying the current ID Template.
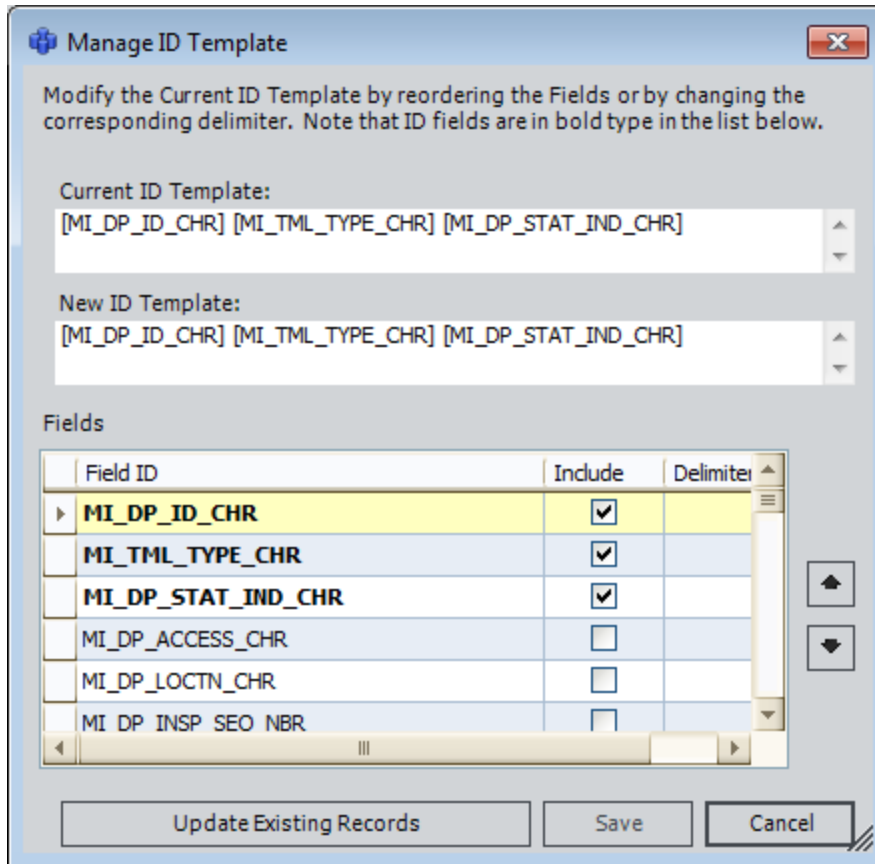
4. Click the **Update Existing Records** button.

   A message appears, explaining that the Meridium APM system will update all existing records with new Record IDs and asking if you want to continue.

   > **Note:** Continuing at this point will cause the Meridium APM system to update existing records based upon the last *saved* ID Template. If you accessed the **Manage ID Template** window, made changes to the ID Template, and did not save them before clicking the **Update Existing Records** button, records will be updated with the ID Template that existed prior to your changes.

5. Click the **Yes**.

   The Meridium APM system updates all existing records in the selected family with new Record IDs that are based upon the updated ID Template. When the update process is complete, the **Manage ID Template** window will close automatically.

Selecting a Family

In many places throughout the Meridium APM Framework application where you can select a family, a hierarchy button, [icon], is available that lets you access the **Select a Family** window, from which you can select a family from a hierarchy of families that are defined in the database. The **Select a Family** window will resemble the following image.
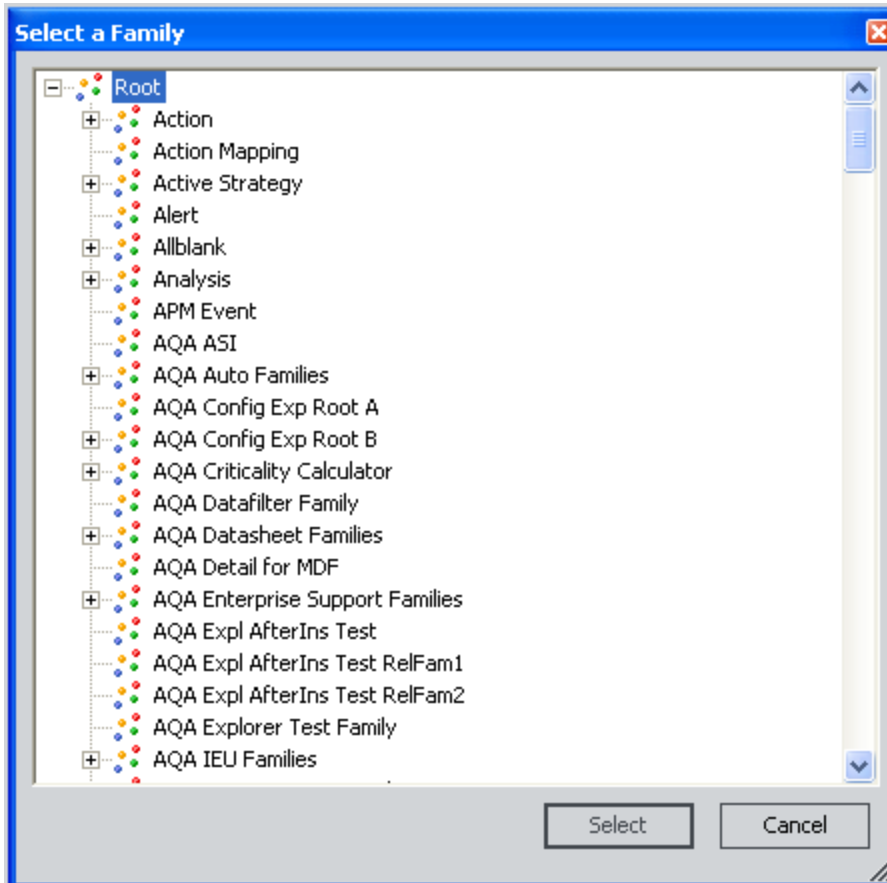
This is only an example of a family hierarchy on the **Select a Family** window. The families that appear in your list will be the ones that are defined in your database.

Note that when the list of families is displayed, you will see only the families to which you have been granted the appropriate privileges via the Configuration Manager. For example, if you are trying to create a new record, the list will display only families to which you have been granted Insert privileges. Likewise, if you are searching for a record, the list will display only families to which you have been granted View privileges.

Examples of where you will see the family hierarchy button are:

- Search results.
- The **New Query** screen of the **Query Builder**.
- The **Calibration Administration** page.

- The **Select Family** dialog box, which appears when you click the **New** button on the toolbar.

To select a family in the **Select a Family** window, expand the hierarchy, highlight a family, and click the **Select** button.

Accessing the Manage Family Fields Window

The **Manage Family Fields** window serves as the central location for viewing and modifying the fields that belong to a given family.

To access the Manage Family Fields window:

1. In the Configuration Manager, in the **Entity Family/Relationship Family** pane, click the **Entity Family** or **Relationship Family** tab.

2. Select the family whose fields you want view.

3. On the **Meridium APM Configuration Manager** main window, in the **Tasks** section, click the **Manage Family Fields** link.

   The **Manage Family Fields** window appears.
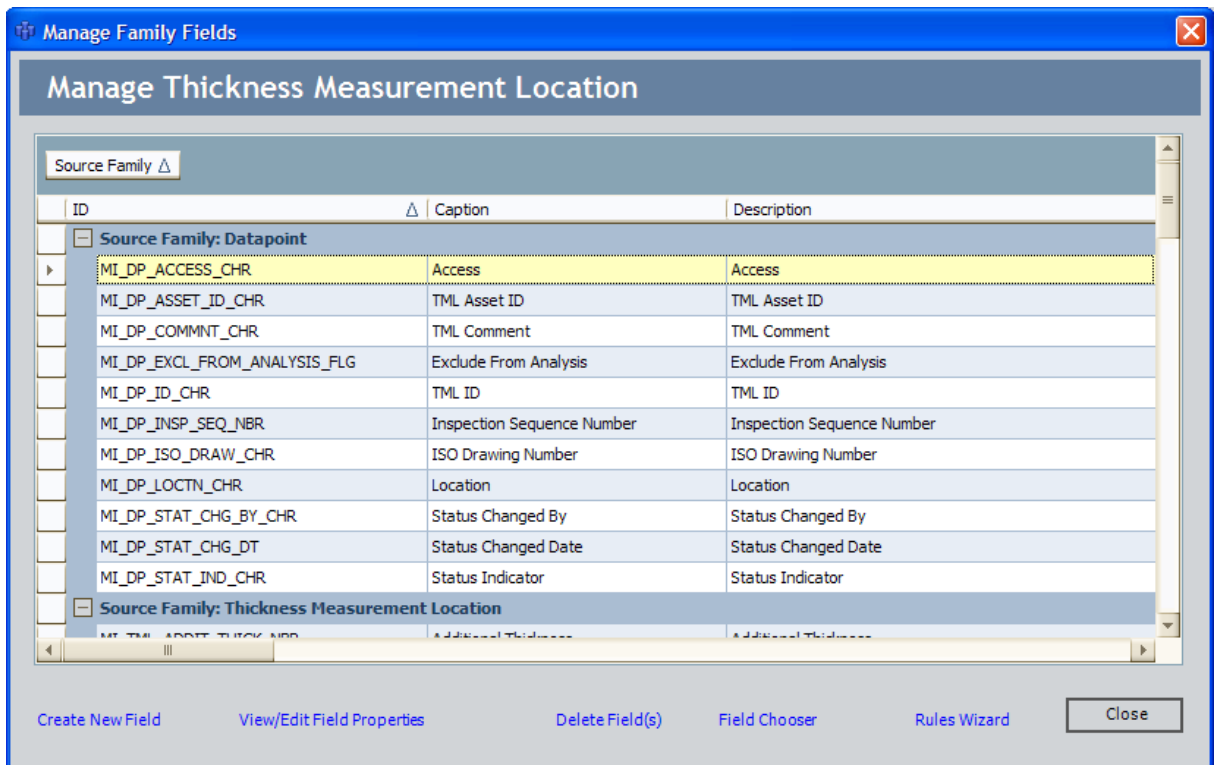
About the Field List and Source Families

The main display area of the **Manage Family Fields** window contains a list of all the fields currently defined for the family. If the fields defined for a given family come from more than one source due to spreading, the list will be divided into multiple sections, each of which corresponds to a source, or base, family.

For example, consider the hierarchy shown in the following image:
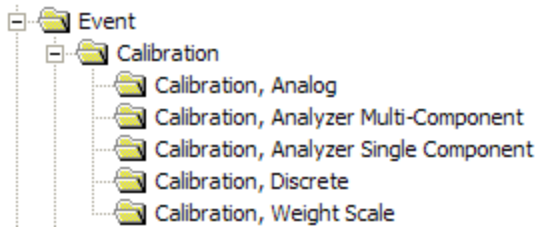
If you access the **Manage Family Fields** window for the Calibration, Analog family in this hierarchy, the list of fields would look something like the one shown in the following image.
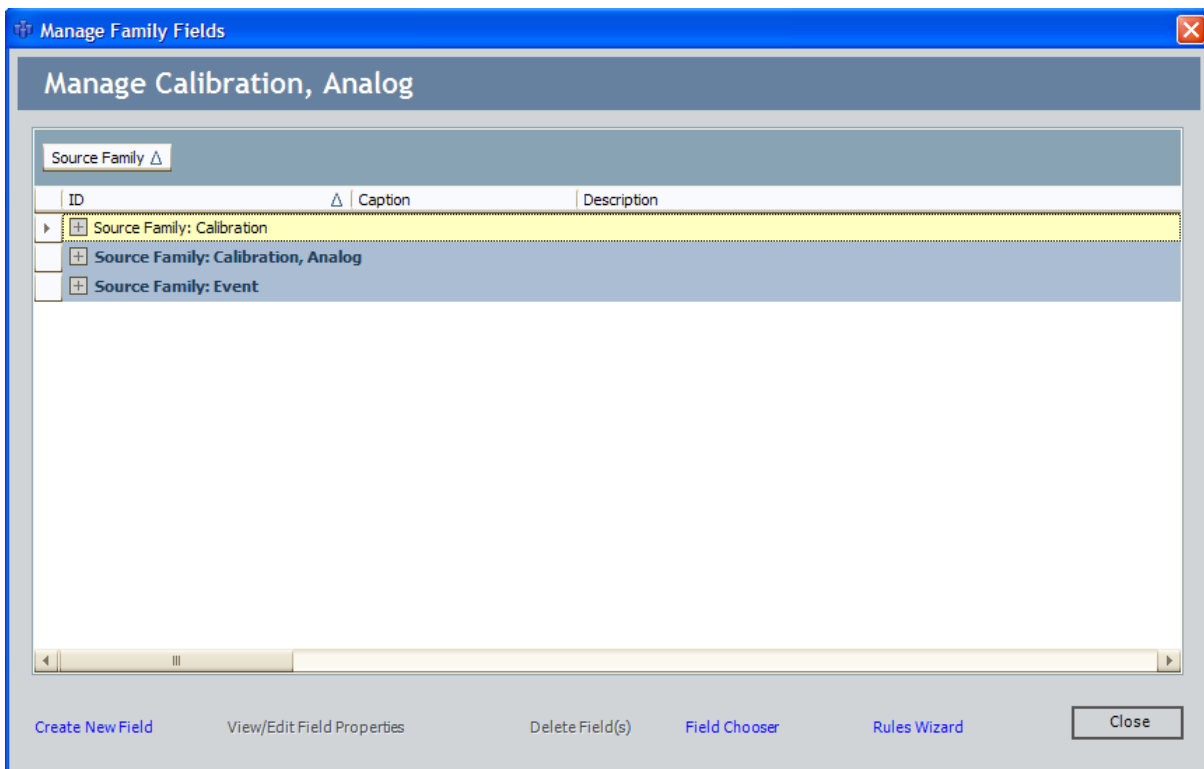
This example shows three sources: *Calibration, Calibration, Analog,* and *Event.* The sources correspond to the Calibration, Analog family's root families and the Calibration, Analog family itself.

You can click the plus sign (+) next to each **Source Family** label to view the specific fields that are defined at that level. The section for the current family (i.e., the family that

appears in the header at the top of the dialog box) is expanded by default when you access the **Manage Family Fields** window.

Columns of Information

The list on the **Manage Family Fields** window is divided into the following columns of information, which you can view using the horizontal scroll bar at the bottom of the list. Each column corresponds to a property that has been defined for the field.

- [ID](#)
- [Caption](#)
- [Description](#)
- [User Help Text](#)
- [ID Field](#)
- Caption Field

> 🛈 **Note:** This column contains a check box that does not correspond to any property that can be configured for family fields on the **View/Edit Field Properties** window. If desired, you can hide this column using the **Runtime Column Customization** option that appears when you right-click the column headers.

- [Spread From Parent](#)

> 🛈 **Note:** On the [**View/Edit Field Properties** window](#), the **Spread to Subfamilies** option lets you spread a field *down* to all subfamilies. In the **Manage Family Fields** grid, however, the **Spread From Parent** flag indicates whether or not a given field has been spread *from* a family.

- [Data Type](#)
- [Formula Field](#)
- [Hyperlink Field](#)
- [Number of Values per Field](#)
- [Required?](#)
- [Validation](#)
- [Valid Values](#)
- [Default Value](#)
- [Disabled](#)
- [Unit of Measure](#)
- [Edit Length](#)
- [Format](#)
- [Physical Column](#)

Links and Buttons

At the bottom of the **Manage Family Fields** window, the following links appear:

- **Create New Field:** Displays the **Create New Field** dialog box, where you can [add a new field to the family](#).
- **View/Edit Field Properties:** Displays the **View/Edit Field Properties** window, where you can [view and modify the properties](#) associated with the selected field.
- **Delete Field(s):** After asking for confirmation, [deletes the field(s) selected in the list](#).
- **Field Chooser:** Displays the **Field Chooser** dialog box, [where you can spread from a family a field that has not already been spread](#). When you select a baseline family that has been delivered with read-only properties, this link is disabled, and you cannot spread fields for the family.
- **Rules Wizard:** Displays the **Rules Wizard**, where you can select one or more fields and [generate field-level rules](#) for all of them.

The **Manage Family Fields** window displays one button: **Close**. You can click the **Close** button to close the **Manage Family Fields** window.

Options for Creating New Fields

You can create new fields for a family in the following ways:

- [Create a new field from scratch](#).
- [Create a new field based on an existing field](#).
- [Add a field using the **Field Chooser**](#).

The instructions in these topics provide details on creating a new field and defining its properties. *After* you complete these instructions:

- You will need to add the new field to the family database table and view by [generating a script to modify the physical database](#).
- If you want the field to be visible in the datasheet so that users can view and edit the values in the field, you will need to [add the field to a datasheet](#).

•

Creating Fields from Scratch

You can create a new, original field from scratch, or you can <u>use an existing field as a starting point, or template, for the new field</u>. These instructions provide details on creating a new field from scratch.

To add a new field to a family:

1. In the Configuration Manager, on the <u>**Manage Family Fields** window</u>, click the **Create New Field** link.

   The **Create New Field** dialog box appears.

2. <u>Configure the properties for the field as desired</u>.

   > ℹ **Note:** Some options are disabled on the **Create New Field** dialog box. Before you can configure any disabled options, you must first save the new field. You can then edit the field to configure the options that were disabled initially.

3. When you are finished configuring field properties, click the **Save** button.

   The new field is saved and appears in the list on the **Manage Family Fields** window.

Creating Fields Based on Existing Fields

Creating a new field based upon an existing field gives you the option of creating a new field without having to define all the properties from scratch. This method is useful if a family will contain two or more similar fields that share some of the same properties.

To create a new field based upon an existing field, you must first modify at least the properties Field ID, Caption, and Physical Column name to be unique values. Because these properties can be modified only on the family where the field physically resides (i.e., they cannot be modified for fields that have been spread from a family), this option cannot be used to copy spread fields. If you want to use this option for creating a new spread field, you must access the family on which the field is actually defined, create a copy of the field, and then spread it to the subfamilies.

> 📄 Note: Creating a field in this way will not copy any field-level rules that exist. Field-level rules will need to be copied over manually. This can be done by editing the family rule project.

To create a new family field based on an existing family field:

1. In the Configuration Manager, on the Manage Family Fields window, select the existing field that you want to use as the template for the new field.

2. Click the **View/Edit Field Properties** link.

   The **View/Edit Field Properties** window appears.

3. Configure the properties as desired. You can modify any property you wish. Before you can save the new field, you must modify at least the following values to make them unique:

- **Field Caption**
- **Field ID**
- **Physical Column**

4. When you have configured all the field properties, click the **Save as New** button.

The new field is created. The **View/Edit Field Properties** window closes, revealing the **Manage Family Fields** window. The newly created field should appear in the list.

Adding Existing Fields Using the Field Chooser

The Field Chooser feature lets you add to a subfamily a field that has been defined for one of its families but has not yet been spread to the subfamily. After you add a field to a subfamily via the Field Chooser, you can view the field properties as you would for any field via the **View/Edit Field Properties** window.

To add a field to a family using the Field Chooser:

1. In the Configuration Manager, on the [Manage Family Fields window](#), click the **Field Chooser** link.

   The **Field Chooser** dialog box appears, displaying the fields defined for the families of the current family that have not yet been spread to that family.
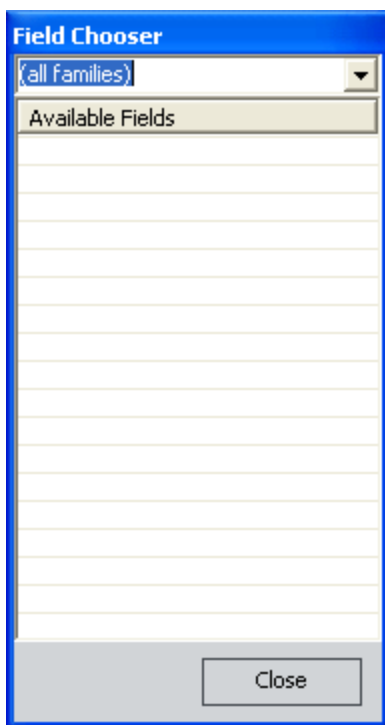
2. Select one more fields from the list, and drag the fields onto the grid area of the **Manage Family Fields** window.

   The field is added to the list.

3. Click the **Close** button.

   The **Field Chooser** dialog box closes.

Formula Fields

A *formula field* is a numeric field that stores a value that has been calculated using rules. To create a formula field, you must:

- Select the **Formula Field** check box on the [View/Edit Field Properties window](#).
- [Define rules for the field to specify how the field value should be calculated](#).

Formula fields differ from non-formula fields in two ways:

- Rather than having a base class of EntityFieldCustomization or RelationshipFieldCustomization, they have a base class of CalculatedEntityFieldCustomization (for entity family fields) or CalculatedRelationshipFieldCustomization (for relationship family fields).
- They contain Formula rules. After a field has been designated a formula field, you can click the **Specify Formula** hyperlink to define the formula that is to be applied to the field. Note that you must select the **Formula Field** check box and *save the field properties* before the **Specify Formula** link will become enabled.

> 🗋 **Note:** Formula rules must be defined manually. They cannot be defined through the Rules Wizard.

When creating a formula field, you should select the **Formula Field** check box and save it *before* creating any rules. This will cause the field's code item to be set up as a formula field. Note that if other rules already exist for a non-formula field and you want to change the field to a formula field, perform the following steps:

- Select the **Formula Field** check box and save the field.
- For entity fields, manually change the base class on the field's code item from EntityFieldCustomization to CalculatedEntityFieldCustomization.
- For relationship fields, manually change the base class on the field's code item from RelationshipFieldCustomization to CalculatedRelationshipFieldCustomization.

> 🗋 **Note:** If you modify the formula for an existing formula field or define a new calculation for a field that was previously not a formula field, the value for that field will not be updated automatically in existing records. The new calculated value will be displayed in formatted query results and in Record Manager. The value stored in the database, however, will not be updated until the records are re-saved. Unformatted queries will display the value that is stored in the database.

Hyperlink Fields

A *hyperlink field* is a field that appears as a hyperlink in Meridium APM datasheets and provides access to an internal or external location defined through a URL. For example, as shown in the following image, the baseline Inspection family contains the fields *Asset ID Link* and *Functional Location ID Link*, both of which are hyperlink fields that provide access to the equipment or location record, as appropriate.

To create a hyperlink field, you must:

- Select the **Hyperlink** check box in the **Identification** section of the **View/Edit Field Properties** window. You can select this check box when you first create the field or at some later time.

  > Note: If you select the **Hyperlink** check box *after* the physical column has been created for the field, you will need to recreate the database views for the change to be applied.

- Define a Default Value rule for the field that specifies the desired hyperlink. The link must be defined using an HTML **<a>** tag that contains an **href** element. The Meridium APM system will interpret the HTML tag as it would be interpreted by any HTML viewer, where:
  - The value defined for the **href** element defines the destination of the link.
  - The text between the opening and closing **<a>** tags defines the text that will be displayed on the datasheet.

For example, the following code would create a hyperlink to the Meridium, Inc. website and display the text Meridium, Inc. website on the datasheet:

Meridium, Inc. website

> **Note:** The double quotation marks in this example are intentional and reflect how quotation marks must be escaped to create a value default value rule.

When creating a hyperlink field, keep in mind that:

- You can use both internal and external URLs as the destination for hyperlink fields.
- Because the destination URL is defined within a default rule, the hyperlink will be the same for ALL records that are created within that family.
- Regardless of the other rules and properties that are defined for the field, the value in a hyperlink field cannot be modified by users via the datasheet.
- A given field can be *either* a formula field or a hyperlink field. It cannot be both.
- Only *character* fields should be used for creating hyperlink fields. Using other field types as hyperlink fields may cause error messages to be displayed when the associated family is accessed in Meridium APM.
- Hyperlink fields can contain only one value. If you set the Number of Values Per Field to more than 1 (one), the value will be ignored.

Multi-Value Fields

Multi-value fields are fields that can contain more than one value. The number of values allowed in a multi-value field is determined by the Number of Values Per Field property, which can be defined for each field on the **View/Edit Field Properties** window. Multi-value value fields can be useful in cases where you want fields to store multiple, yet distinct, values.

Consider, for example, that you use the *Work Order* family to store records containing information about work performed on equipment in your plant. Now, assume that the Work Order family contains the *Maintenance Type* field, which is meant to identify the type of work represented by a given Work Order record. In this case, it may not be adequate to associate a *single* maintenance activity with each Work Order record. For instance, some work orders may require that a piece of equipment be repaired and cleaned. If more than one activity is performed against a single piece of equipment by the same person at any given time, you may want to allow multiple values to be selected in the Maintenance Type field. In this way, users would be able to associate more than one activity with each Work Order record rather than creating multiple Work Order records.

To implement this behavior, you would:

1. On the **View/Edit Field Properties window** for the Maintenance Type field, in the **Number of Values Per Field** text box, type the number of values that you want to allow users to specify. For example, if you wanted to allow up to three maintenance types, you would type **3**.

   > 🗋 **Note:** By default, the only validation performed on multi-value fields is that which limits users from specifying *more than* the number of values allowed. By default users can specify no value, one value, or any other number of values up to the limit specified by the Number of Values Per Field property.

2. Create a Valid Values rule to populate the Maintenance Type field with a list of valid maintenance activities.

When users access the Work Order family in the Meridium APM Framework application, the Maintenance Type field will display the list of allowed values. A check box will appear to the left of each value. Users can select the check box for each desired value, as shown in the following image.

The selections that are made will appear as a comma-delimited list in the Maintenance Type field, as shown in the following image.

This is just one example of how you might implement a multi-value field. You are not required to create Valid Values rules for multi-value fields. If there is no Valid Values rule defined for a multi-value field, the datasheet will display a drop-down list where users can type (or select in the case of date fields) the values that they want to add to the field, as shown in the following image.

When working with multi-value fields in the Configuration Manager, note the following:

- Character, date, and numeric fields can be configured to contain more than one value. Numeric and date fields behave in a way similar to the example provided above. Binary, text, and logical fields can not be configured as multi-value fields.

- When you first create a date or numeric field and specify a value other than 0 (zero) in the **Number of Values Per Field** text box, the value will not be saved. After you save the field, you will need to access the **View/Edit Field Properties** window and specify the desired number of values again. When you save your change the second time, the setting will be saved. This behavior applies only to numeric and date fields. Character fields will retain their **Number of Values Per Field** property when they are initially saved.

- You cannot create a Default Value rule for a multi-value numeric or date fields to populate the field with a default literal value. Doing so will cause an error message to be displayed when a user accesses the associated family in the Meridium APM Framework application.

About Field Properties

Field properties can be configured on the **Create New Field** dialog box when you are creating a new field and on the **View/Edit Field Properties** window when you are modifying an existing field or creating a new field based on an existing field.
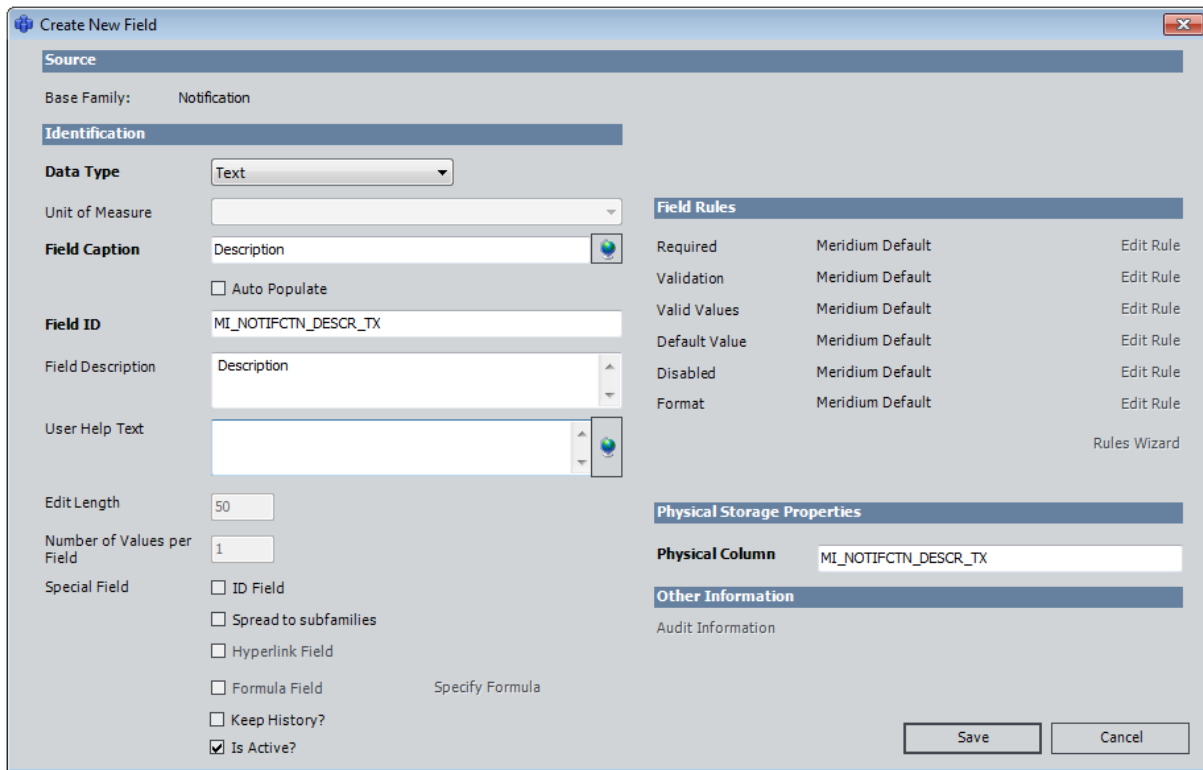
On the **Create New Field** dialog box and the **View/Edit Field Properties** window, you can perform the following tasks for a given family field:

- View source information.
- Configure identification properties.
- Configure spread properties.
- Manage field-level rules.
- Configure physical storage settings.

Accessing Field Properties

From the **Manage Family Fields** window, you can access the properties associated with any field that belongs to the current family.

To access the properties of an existing field:

1. In the Configuration Manager, on the **Manage Family Fields window**, select the field whose properties you want to access.

2. Click the **View/Edit Field Properties** link.

   The **View/Edit Field Properties** window appears.

Modifying Field Properties

After you have created a field for a family, you can view and modify its properties on the View/Edit Field Properties window. When modifying field properties, keep in mind that:

- Depending upon whether or not a given field has been spread down to the current family from a higher-level family, some field properties may not be editable.
- Code items that store field-level rules for a given field are name based upon field IDs. If you modify the field ID for a field, a NEW code item will be created using the update field ID. The code item that corresponds to the OLD field ID will not be deleted, and any rule code that exists within the old code item will not be copied to the new code item automatically.
- Some attributes of baseline fields cannot be modified. If you attempt to make a modification that is not allowed, an error message will be displayed, and your change will not be saved.
- Some baseline fields cannot be modified because they belong to a family that has been delivered with read-only properties. When you view the field properties for a field that meet this criteria, some field properties will not be editable.

To modify the properties of a field:

1. In the Configuration Manager, access the View/Edit Field Properties window for the desired field.

2. Modify the field properties as desired.

3. Click the **Save** button.

   Your changes are saved to the database.

Viewing Source Information

The **Source** section is read-only and displays the name of the **Base Family**, which indicates where the field originated, or is physically located.

If you are viewing a field for a subfamily, the field may have been spread down from a family. For example, consider a hierarchy that looks like the one shown in the following image.
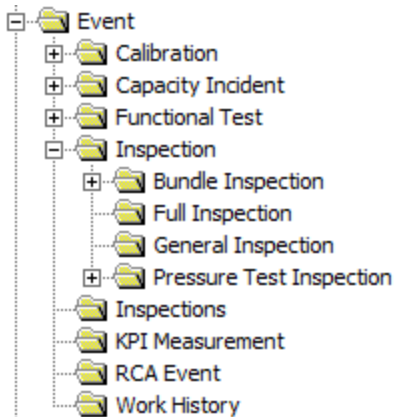
Now, suppose the Event ID field is defined for the Event family and is spread to each subfamily. In this case, you could view the field properties for the Event ID field at the Event and Calibration levels. But no matter the level at which you view the field, the Base Family would always be the same: Event.

•

Available Field Identification Properties

The **Identification** section of **View/Edit Field Properties window** contains identifying information and rules for a given field.

In the **Identification** section, you can define the properties described in the following table.

Property	Description
Data Type	Identifies the type of data that will be stored in the field. The data type controls which other properties are enabled or disabled, as appropriate for that type of data. You can choose from the following types: • **Binary:** The field can contain binary data, which is a coding system that represents data using a series of numbers. Binary fields cannot be displayed on datasheets. • **Character:** The field can contain any combination of characters up to the limit specified by the value in the **Edit Length** text box. On datasheets, character fields appear as text boxes, into which users can type values, or drop-down lists, from which users can select a valid value. • **Date:** The field can contain date and time data. On datasheets, date fields display a text box into which users can type a date and time. The pop-up Calendar is also available on datasheets and allows users to select a date and enter the desired time. • **Number:** The field can contain numeric data. On datasheets, numeric fields appear as text boxes, into which users can type values, or drop-down lists, from which users can select a valid value. • **Text:** The field can contain any combination of characters with no limit. On datasheets, text fields appear as text boxes into which users can type the desired value. From text fields, users can also access the text editor, which provides more space for typing data and offers a spell checking feature. • **Logical:** The field can contain a value that represents the equivalent of *True* or *False*. On datasheets, logical fields appear as check boxes. Users can select the check box to specify a value of *True* or clear the check box to specify a value of *False*. Note that if you create a new logical field for a family that already contains records, the field will be set to *Null* instead of *False* in all the existing records.

Property	Description
UTC?	Indicates that the date field will *store* time using the Coordinated Universal Time (UTC) format. You can select this check box to indicate that you want date values to be stored in UTC time in the database. On the interface, the date value will be displayed using the *local time for that user*, which is determined by the value stored in the Timezone field in their Security User record.
	The **UTC?** check box appears only if the data type is set to *Date*. This check box is disabled if the field was spread from a parent family to the family that you are currently working with. Some baseline fields are delivered with this property set to *True.* These fields are documented in context of the modules that use them.
	You can set this value to *True* only in Date fields belonging to families for which records do *not* already exist. If records exist in the family to which the field belongs, and you still want to change the UTC value to *True*, you will need to contact Meridium, Inc. for help in assessing the impact of modifying the UTC field property and performing any additional steps that are needed as a result.
Unit of Measure	Identifies the unit of measure (UOM) associated with the values stored in a numeric field. You must select a UOM setting whenever the data type property is set to *Number*. You can, however, set the Unit of Measure to *None*, which means that no unit of measure will be associated with the field. The **Unit of Measure** list is populated with the UOMs that have been defined in the database.
	Note that:
	• The unit of measure (UOM) that you specify here designates the UOM in which values will be *stored* in the Meridium APM database. This is not necessarily the same as the unit of measure in which it will appear to users in the Meridium APM Framework application or the Meridium APM Web Framework.
	• You can change the unit of measure for existing fields only if there are no values stored in that field in any existing records.
	• For fields that have been spread to a subfamily from a higher-level family, the UOM is determined by the *source* field. You cannot define a different UOM at the subfamily level even when field properties are set to be changed at the subfamily level. If you modify this value at the subfamily level, your changes will not be saved.

Property	Description
Field Caption[1]	Specifies the label for the field as it will appear in various places throughout Meridium APM. To define a field caption, type a name in the **Field Caption** text box. A field caption is required for all fields and can be translated. Each field caption must be unique within its family. Note that defining translations for field captions alone will not ensure that translated strings appear everywhere throughout the Meridium APM Framework application or the Meridium APM Web Framework. If you want to use translated strings, you should define them both for field captions *and* for datasheet captions. If the **Auto Populate** check box is selected, the value that you type in the **Field Caption** text box will be used to populate the **Field ID** and **Field Description** text boxes automatically. If the **Auto Populate** check box is not selected, the field ID and field description can be defined independently of the field caption.

Property	Description
Field ID	Specifies the unique ID that identifies the field. The field ID is required for *all* fields and must be unique. After a field exists, we recommend that you not modify the field ID. If you need to change the field ID, you should delete the existing field and recreate it from scratch.
	You can optionally set the field ID automatically based upon the field caption.
	• If the **Auto Populate** check box is selected when the field is first created, the **Field ID** text box will be disabled and populated automatically based upon the value that is typed in the **Field Caption** text box.
	• If the **Auto Populate** check box is not selected when the field is first created, the **Field ID** text box will be enabled, and the field ID can be defined independently of the field caption.
	In new or existing fields where the **Auto Populate** check box is not selected, selecting the **Auto Populate** check box will cause the existing field ID to be overwritten.
	If you choose not to populate the field ID automatically and instead define it yourself, make sure that the field ID is different from the family ID of the family to which it belongs. Because the code item names in family rule projects are based upon family and field IDs, if the field ID matches the family ID, the Meridium APM system will not be able to create a unique code item for both the family and the field. When the field-level code item is created, it will overwrite the family-level code item. In order for the code items to be unique, the IDs must be unique.
Field Description	Specifies an optional, textual description of the field and its function. The field description is optional. If the **Auto Populate** check box is selected, the **Field Description** text box will be populated automatically based upon the value that is typed in the **Field Caption** text box. In new or existing fields where the **Auto Populate** check box is not selected, selecting the **Auto Populate** check box will cause the existing field ID to be overwritten.

Property	Description
User Help Text	Specifies the explanation of the field, which will be displayed to users when they are entering values in that field via the Meridium APM Framework application. In the Meridium APM Framework, when a user pauses on a field, the user help text will appear in a tooltip. The user help text can be especially useful if the field purpose is complex. You may also want to use the user help text property to describe how a field will behave as a result of the rules that are defined for it. The tooltip can display at one time the entire help text string that is defined in the **User Help Text** text box. We recommend, however, that you limit user help text to one sentence because, while the text will wrap automatically within the tooltip, depending upon the length of the user help text, the tooltip could exceed the width of the Meridium APM window and your computer screen, which will cause the user help text to be cut off.
Edit Length	Specifies the maximum number of characters that will be accepted for the field value. This setting applies to character fields only. Text fields have no limit. All other fields have a limit of 50 characters. The default setting for character fields is 50, but you can type any value up to 2000 (the maximum number of characters allowed for a character field). We recommend that you not modify the field lengths of baseline Meridium APM family fields.
Number of Values per Field	Specifies the number of values that can be entered into a field. The default value is 1 (one), but you can specify any number up to 2000. Note that: • When you create a numeric or date field and modify the value in the **Number of Values per Field** text box, your changes will not be saved when you save the field. If you reopen the **View/Edit Field Properties** window, you will see that the value has been rest to 1 (one). If you change the value a second time, it *will* be saved. • You cannot modify this property for existing fields that belong to a family for which at least one record exists. After at least one record exists in a given family, this property is read-only. If you want to change this value for an existing field, you will need to delete the field and create a new one. • Via the Configuration Manager, you can choose the background color that will appear on datasheets for multi-value fields.

Property	Description
ID Field	Designates the field as an ID field. All fields that are designated as ID fields will be spread automatically to all subfamilies of the current family. Note that when you select the **ID Field** check box, the **Spread to Subfamilies** check box will be selected automatically to indicate that the field will be spread to subfamilies automatically. This setting cannot be modified for fields that have been spread *from* higher-level families. We recommend that you select this check box for any field that you plan to use in the ID Template for this family. ID fields will appear in bold on the **Manage ID Templates** dialog box.
Spread to subfamilies	Specifies whether or not the field will be spread to subfamilies of the current family, meaning that the field will exist in all subfamilies of the current family. If you want to spread the field, select this check box. Selecting the **Spread to subfamilies** check box will cause this field to be spread to *all* subfamilies automatically. You cannot undo spreading for specific subfamilies. If you know that you will not need a field on all subfamilies, instead of selecting the **Spread to sub-families** check box at the root level, you can spread fields to individual subfamilies using the **Field Chooser** feature. If you select this option on a field for which physical storage has not yet been created, when you save the field, an error message appears, indicating that the field was saved but that the views were not successfully recreated. This message does not indicate a problem and is to be expected. It is not possible to recreate views until the physical storage has been created in the database. To resolve this issue, simply create the physical storage for the field. This setting cannot be modified for fields that have been spread *from* higher-level families.
Hyperlink Field	Allows you to define the field as a hyperlink field. In addition to selecting this check box, you will need to define a rule to create a fully functional hyperlink field. This setting cannot be modified for fields that have been spread *from* higher-level families.
Formula Field	Allows you to define the field as a formula field. Fields can be designated as formula fields only after they have been saved; the **Formula** check box is disabled when you first create a new field. This setting cannot be modified for fields that have been spread *from* higher-level families.

Property	Description
Keep History?	Specifies whether or not revision history will be saved for the field. If you select this option, when a change is made to the value in a field, a copy of it is created and saved to a history log. You can view the revision history in the Record Manager in the Meridium APM Framework application. This setting cannot be modified for fields that have been spread *from* higher-level families.

Is Active?	Determines whether or not the field is active. This setting cannot be modified for fields that have been spread *from* higher-level families. Inactive fields will not appear in the following locations:
	
	• In the **Datasheet Builder**, in the **Editing <Datasheet ID>** section, in the list of available fields in the **Values** column. If a field is flagged as inactive after it already exists on a datasheet, it will not be removed from the **Values** column automatically. When a user views an inactive field on a datasheet, it will be disabled, and any value stored in that field will not be displayed.
	
	• In the query design, in the list of available fields.
	
	    ▪ Inactive fields will not appear in the query source lists, but you can optionally make them appear if desired.
	
	    ▪ Inactive fields will not appear in the **Fields** list. There is no way to make them appear.
	
	If a field is flagged as inactive after it is already being used in a query, it will continue to be included in the query unless you remove it manually.

Spreading Changes to Captions, Descriptions, and Help Text

In some cases, you may want to spread the attributes of a field in a family to one or more subfamilies but make the Caption, Description, and Help Text unique to the sub-family. To do this, you can enable spreading at the root level and then modify the Caption, Description, and Help Text at the subfamily level as needed.

> ⓘ **Hint:** You can modify the Caption, Description, and Help Text even when the Spreading attribute of the subfield is set to **Keep all properties of this field the same as they are in the [Parent] family**. All other properties will be read-only at the sublevel and inherited from the source family.

Because the Caption, Description, and Help Text can be customized at the subfamily level, whenever you make a change to one of these properties at the *root* level, you will need to decide whether or not you want to spread it to all sublevels. In general:

- You will *want* to spread changes from the root level to the subfamily if the sub-family has not been customized.
- You will *not want* to spread changes from the root level to the subfamily if the sub-family has been customized.

This choice is facilitated by the following dialog box, which appears whenever you modify the Caption, Description, or Help Text of a root field that has been spread to one or more subfields. This dialog box will appear when you modify the default, non-localized value or any localized value.
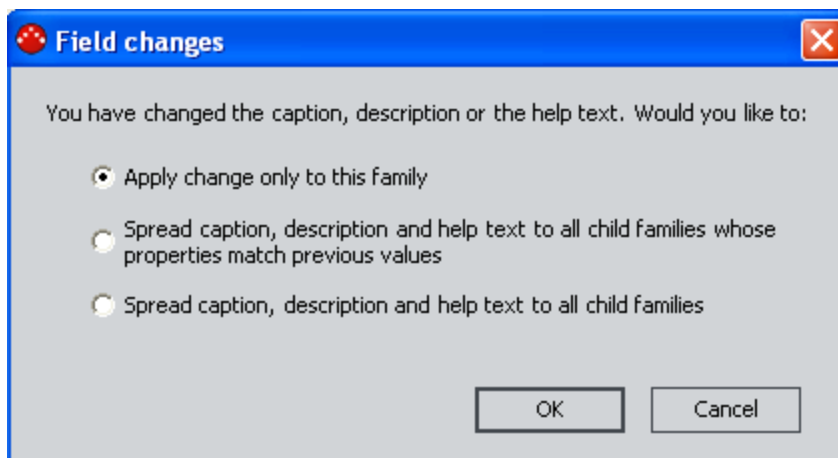
You can choose from the following options:

- **Apply change only to this family:** Any change to the Caption, Description, or Help Text will be applied only to the family. The properties will not be updated in the subfamilies.
- **Spread caption, description, and help text to all child families who properties match previous values:** The Caption, Description, and Help Text will be spread to

any subfamilies whose Caption, Description, or Help Text matched the value in the family *before* the change was made.

> ⓘ **Note:** When you select this option, the comparison is performed using only the default, non-localized value. If the default value of the subfamily matches the pre-change value in the family, the subfamily will be updated with the default, non-localized value and the entire set of localized values defined for the family.

- **Spread caption, description, and help text to all child families:** The Caption, Description, and Help Text will be applied to ALL subfamilies, regardless of their current values. In other words, the values in all subfamilies will be overwritten.

> ⚠ **IMPORTANT:** If you select this option, the Caption, Description, and Help Text will be spread from the family to all subfamilies, regardless of whether all three values have actually been modified. Note also that this option will spread both the default, non-localized value and all localized values.

Configuring Spreading Properties

The **Spreading** section of the **View/Edit Field Properties window** allows you to define the source of Special properties and rules for fields that have been spread to the current family from any higher-level family.

> ℹ️ Note: The **Spreading** section appears on the **View/Edit Field Properties** window only when you are viewing the properties of a field that has been spread to a sub-family from a higher-level family. This section does not appear on the **Create New Field** dialog box.

The **Spreading** section displays two options:

- **Keep all properties of this field the same as they are in the [Source] family:** Specifies that the field-level rules and Special properties will be the same for this field as they are defined in the family, where [Source] is the name of the family from which the field has been spread.

- **Change the properties of this field for [Subfamily]:** Specifies that the field can have its own rules and Special properties that are different from the ones defined at the level from which the field was spread.

  > ℹ️ Note: When you select this option, the Meridium APM system will create a code item for the spread field in the family rule project of the *subfamily*.

Working with the Field Rules Area

The **Field Rules** section of the <u>View/Edit Field Properties</u> <u>window</u> displays information about the <u>field-level rules</u> have been defined for a field and provides you with access to tools that let you modify the rules.

The **Field Rules** area of the **View/Edit Field Properties** window reflects the information from the *most recently compiled instance* of the family rule project. The **Field Rules** section displays the following columns of information:

- The first column displays the rule type: <u>Required</u>, <u>Validation</u>, <u>Valid Values</u>, <u>Default Value</u>, <u>Disabled</u>, and <u>Format</u>.
- The second column displays the current setting for each rule type. This column will display one of the following values:
    - <u>Meridium APM Default</u>: Indicates that the baseline Meridium APM rules will be used.
    - **Custom:** Indicates that custom rules exist for the field. These rules may be defined within the field class itself, inherited from the Client area of the <u>Rules Library</u>, or inherited from another field customization.
- The third column displays the **Edit Rule** link, which you can click to launch the Meridium APM Rules Editor, where you can view and modify the custom rule code. This link is enabled only for rule types that are designated as *Custom* in the second column.

At the bottom, right of the **Field Rules** section, the **Rules Wizard** link is displayed and provides access to the Rules Wizard, which will take you step-by-step through the process of defining standard field-level rules without having to access the rule code itself.

> ⓘ **Note:** If you are viewing a field that has been spread down from a higher-level family, you will be able to modify the **Field Rules** section only if the option **Change the properties of this field for [Child Family]** is selected in the **Spreading** section. Otherwise, all field rules will be the same as those defined for the field in the family.

Configuring Physical Storage Properties

The **Physical Storage Properties** section displays information about where the field is physically stored in the database.

The **Physical Column** text box, which appears in the **Physical Storage Properties** section, contains a value that specifies the name of the physical table column in which the field is stored. You can type a value in this text box, or you can select the **Auto Populate** check box (below the **Field Caption** text box in the **Identification** section) to populate this value automatically using a combination of the physical table name, segmented caption, and data type extension. The **Physical Column** setting is required for all fields and must be unique.

> ⓘ **Note:** For fields that have been spread down to a subfamily from a higher-level family, the physical column name is determined by the value for the source family. For spread fields, you cannot modify this value at the subfamily level. If you change the value in the Physical Column text box at the subfamily level, your changes will not be saved.

Viewing the Audit Information for a Field

The **Other Information** section of the [View/Edit Field Properties window](#) displays a link that lets you view the audit information for the field. On the **Create New Field** dialog box, this link is disabled as no audit information exists for new fields.

To view the audit information for a field:

- In the Configuration Manager, on the **View/Edit Field Properties** window, click the **Audit Information** link.

  The **Audit Information** dialog box appears, showing when the field was last modified and who modified it.
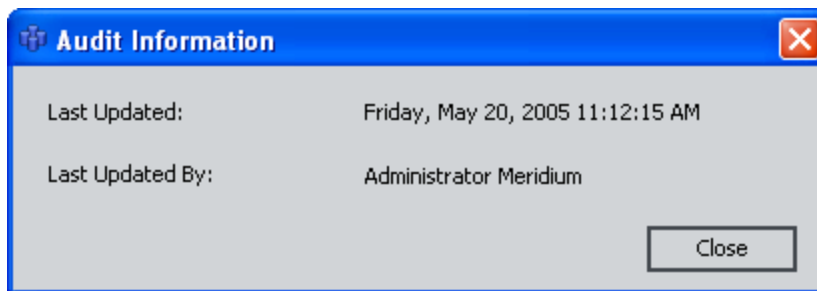
About Field Sequence Numbers

All the fields within a family are assigned a field sequence number, which determines the order in which fields will be displayed, loaded, and processed during certain operations. The default field sequence number is assigned automatically by the Meridium APM system when a field is created, but you can change the field sequence for a family, if desired.

In some cases, field sequence number are inconsequential and do not impact how the system functions. In other cases, particularly in cases where rules are fired, field sequence numbers are important. To make sure that these operations do not result in errors, you should define field sequence numbers for field as appropriate for the rules that exist for that family.

For example, consider a family that contains fields **A**, **B**, and **C**, each of which is formatted to display a list of valid values. Now, suppose that rules have been written for the field to specify that the value selected in list **A** determines the available values in list **B**, and the value selected in list **B** determines the value selected in list **C**. On the datasheet, where you have control over the order in which fields are displayed, you can specify that list **A** should appear first, then list **B**, and finally list **C**. This would encourage users to select a value first from list **A**, then from list **B**, and finally list **C**, causing all the rules to be fired in the intended order.

In operations where you do not have control over the order in which fields are processed, however, you must rely upon field sequence numbers. In this example, you would want to set the field sequence numbers such that field **A** would be processed first, followed by **B**, and then by **C**.

Field sequence numbers are used when fields are processed during an import or export.

> ⓘ **Note:** Only one field sequence number can be defined for each field. This means that you will need to define field sequence numbers that are appropriate for visual and functional purposes. If you modify field sequence numbers to adjust a certain display, keep in mind the impact of those changes on the loading and processing of data.

Note that due to a current known issue, whenever you modify a field in a family, the Meridium APM system resets the field sequence for that field. Therefore, you should wait to set the field sequence for a family *after* you have set up and configured all fields in the family. If you do need to modify a field after setting the field sequence, you will need to modify the field sequence after modifying the field.

Modifying Field Sequence Numbers

The **Family Fields Sequence Order** window contains a list that displays all the fields in a given family. The list does not include the field sequence numbers themselves. Instead, the order in which the fields are listed corresponds to the sequence number defined for each field.

When you click the **Save** button on the **Family Fields Sequence Order** window, the fields will be assigned a sequence number that corresponds to the order in which they are displayed in the list on the **Family Fields Sequence Order** window. The first field in the list will be assigned a 1, the second field will be assigned a 2, the third field will be assigned a 3, and so on. You can reorganize fields within the list to determine the sequence number that will be assigned to each field. Note that the actual field sequence number values are unimportant. What matters is the order of each sequence number relative to the sequence numbers defined for other fields in the same family.

To access the Family Fields Sequence Order window:

1. In the Configuration Manager, in the **Entity Family/Relationship Family** pane, select the family whose field sequence you want to view.

2. In the **Tasks** section, click the **Sequence Family Fields** link.

   The **Family Fields Sequence Order** window appears.
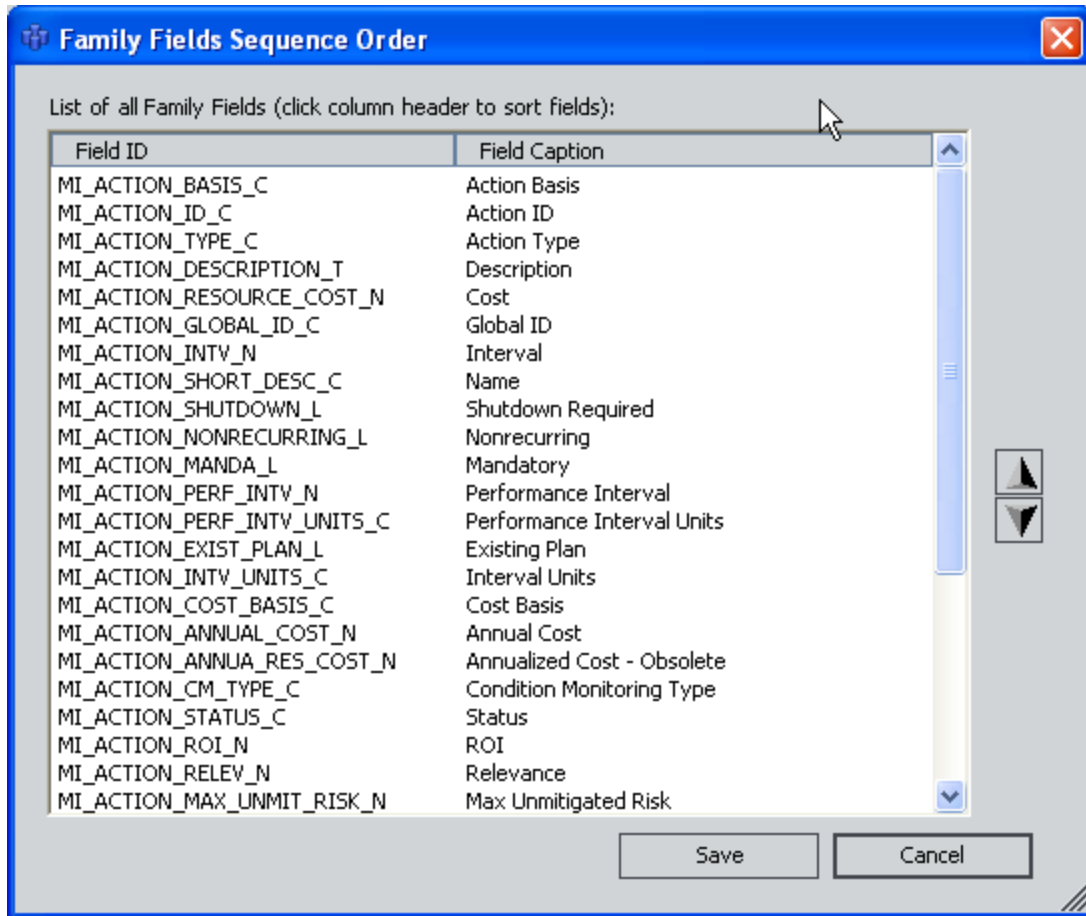
The order of the fields in the list serves as an exact representation of the field sequence. Note that the list contains all fields that exist for the selected family, including any fields that have been spread from a family.

3. In the list of fields, select the field(s) whose order you want to change.

4. Click ▲ to move the field(s) up in the list.

   -or-

   Click ▼ to move the field(s) down in the list.

5. Move any additional fields you wish.

6. When you are finished reordering the fields, click the **Save** button.

   Your changes are saved.

About Field Sequence Numbers in Imports and Exports

When you export a family using the Import/Export Metadata Utility, the sequence number for each field is included in the FIELD_SEQUENCE_NBR column on the **Fields** tab of the Microsoft Excel file, as shown in the following image.

	A	B	C	D	E	F	G	H	I	J	K	L
1	FIELD_ID	FIELD_FA	FIELD_IS	FIELD_SEQUENCE_NBR	FIELD_PF	FIELD_CA	FIELD_DE	FIELD_DA	FIELD_LE	FIELD_UN	FIELD_IS	FIELD_
2	ASSET_IC	Motor	True	1	ASSET_IC	Asset ID<	Asset ID	Character	40		True	False
3	ASSET_D	Motor	True	3	ASSET_D	Asset Des	Asset Des	Character	150		True	False
4	ASSET_M	Motor	True	4	ASSET_M	Asset Mar	Asset Mar	Character	50		False	False
5	ASSET_S	Motor	True	5	ASSET_S	Asset Seri	Asset Seri	Character	50		False	False
6	ASSET_M	Motor	True	6	ASSET_M	Asset Moc	Asset Moc	Character	50		False	False
7	ASSET_IN	Motor	True	7	ASSET_IN	Asset Inst	Asset Inst	Date	0		False	False
8	ASSET_S	Motor	True	8	ASSET_S	Spared ?<	Spared ?	Boolean	1		False	False
9	ASSET_C	Motor	True	9	ASSET_C	Critical As	Critical As	Boolean	1		False	False
10	ASSET_F	Motor	True	10	ASSET_F	Material H	Material H	Character	50		False	False
11	ASSET_S	Motor	True	11	ASSET_S	Asset Stat	Asset Stat	Character	20		False	False
12	ASSET_S	Motor	True	12	ASSET_S	Asset Ser	Asset Ser	Character	50		False	False
13	ASSET_A	Motor	True	13	ASSET_A	Asset Add	Asset Add	Text	0		False	False
14	ASSET_D	Motor	True	14	ASSET_D	Asset P&I	Asset P&I	Character	30		False	False

Information / Families / FamilyDataFilters / FamilyReports \ Fields /

In addition, the **Families** tab of the export file includes the FAMILY_APPLY_FIELD_SEQUENCES column, which contains a setting that determines how field sequence numbers will be handled for each family included in the file during an import. When the FAMILY_APPLY_FIELD_SEQUENCES column is set to:

- **False** (the default setting for all export files created by the Meridium APM system), when the file is imported, if the family already exists in the target database, the existing field sequence numbers will not be overwritten with the field sequence numbers in the import file.

- **True**, when the file is imported, if the family already exists in the target database, the existing field sequence numbers *will be* overwritten with the field sequence numbers in the import file.

If you import a file that contains information for a family that does not already exist in the target database, the field sequence numbers that are defined in the import file will be applied to the fields in the new family, regardless of the value in the FAMILY_APPLY_FIELD_SEQUENCES column.

> ⓘ **Note:** The FAMILY_APPLY_FIELD_SEQUENCES exists only in export files that were created using V3.2.3 or later. If desired, you can add the column FAMILY_APPLY_FIELD_SEQUENCES to any file that does not already contain it to define the action that should be taken on import. If this column does not exist in the file, when you import the file and the family already exists in the target database, a value of **False** will be assumed (i.e., the existing field sequence numbers will not be modified), and a warning message will be written to the import log.

Note that Meridium APM enforces no restrictions on duplicate field sequence numbers. If you import a file that contains a subset of fields that are defined for an existing family and any of the field sequence numbers are the same as those defined for existing fields,

the import will be successful and will result in duplicate field sequence numbers within the family. Additionally, Meridium APM allows duplicate field sequence numbers for different fields within the same import file. Duplicate field sequence numbers may cause rules to function improperly and should be corrected via the **Family Fields Sequence Order** feature, which will reset all sequence numbers to reflect the order of the fields as displayed on the **Family Fields Sequence Order** window.

Note also that the features described above will also exist in an XML export file if you choose to use that format instead. We recommend, however, that you use Microsoft Excel files whenever you plan to review or modify values since the Microsoft Excel format is easier to use.

Deleting Fields from a Family

Deleting a family field removes the metadata definition from that family and from all subfamilies to which it has been spread. Note that deleting a family field via the **Manage Family Fields** window will *not*:

- Delete the physical column from the table in the database. To delete the physical column from the database, you must generate a script to modify the family's physical table. This means that the physical deletion from the database does not occur when you complete the steps outlined below. If you accidentally delete a field, the data can be recovered up until the point that the physical column is deleted from the database.

- Delete the associated code item from the family rule project if the code item contains logic. If you want to delete the code item, you will need to do so manually via the **Meridium APM Rules Editor** (VSTA). If the code item exists but does not contain any logic, it will be deleted when you delete the field via the **Manage Family Fields** window.

- Delete the field from datasheets. The field must be deleted from all datasheets manually, including datasheets defined for the family and any datasheets that exist for the subfamilies to which the field has been spread.

Some baseline fields cannot be deleted because they belong to a family that has been delivered with read-only properties. When you try to delete a field that meets this criteria, the **Delete Field(s)** link will be disabled.

To delete a field from a family:

1. On the **Manage Family Fields** window, select the field(s) that you want to delete.

2. Click the **Delete Field(s)** link.

   A confirmation message appears, asking if you really want to delete the field(s).

3. Click the **Yes** button.

   If the field has been spread to subfamilies, a second confirmation message appears, asking if you want to delete the field from those subfamilies.

4. Click the **Yes** button. If you selected multiple fields, you can click the **Yes to All** button to apply this response to all the fields that you selected. You can click the **No** button to cancel the deletion.

   > ⓘ **Note:** Clicking the **Yes** button will delete the field from the current family and *all* subfamilies.

   If data or field history exists for the field, a third confirmation message appears, warning you that data exists and asking if you still want to delete the field. When the field is deleted, *all* data will be lost for this field for records in the current family and records in any subfamily (if the field has been spread).

5. Click the **Yes** button. If you selected multiple fields, you can click the **Yes to All** button to apply this response to all the fields that you selected. If you want to preserve the data, you can click the **No** button to cancel the deletion.

The fields are deleted.

About Managing Relationship Definitions

Via the Configuration Manager, you can manage relationship definitions by:

- Using the **Relationship Definition Wizard** to [create all possible relationships between families](#).
- [Creating a relationship family manually](#).
- [Deleting predecessor families](#).
- [Deleting successor families](#).
- Using relationships to enable specific functionality, such as Reference Documents functionality.

Accessing the Relationship Definitions Window

The **Relationship Definitions** window is where you can manage all the relationship definitions for a given relationship family.

To access the Relationship Definitions window:

1. In the Configuration Manager, in the **Entity Family/Relationship Family** pane, click the **Relationship Family** tab.

2. In the list of relationship families, select the relationship family for which you want to configure relationship definitions.

3. In the **Tasks** section, click the **Manage Relationship Definitions** link.

   The **Relationship Definitions** window appears.

Relationship Definitions

Relationship Definition Wizard

Predecessor(s)

Cardinality

- ● Zero or One
- ○ Zero or Many

Add   Delete

Successor(s)

Cardinality

- ○ Zero or One
- ● Zero or Many

Add   Delete

Save   Close
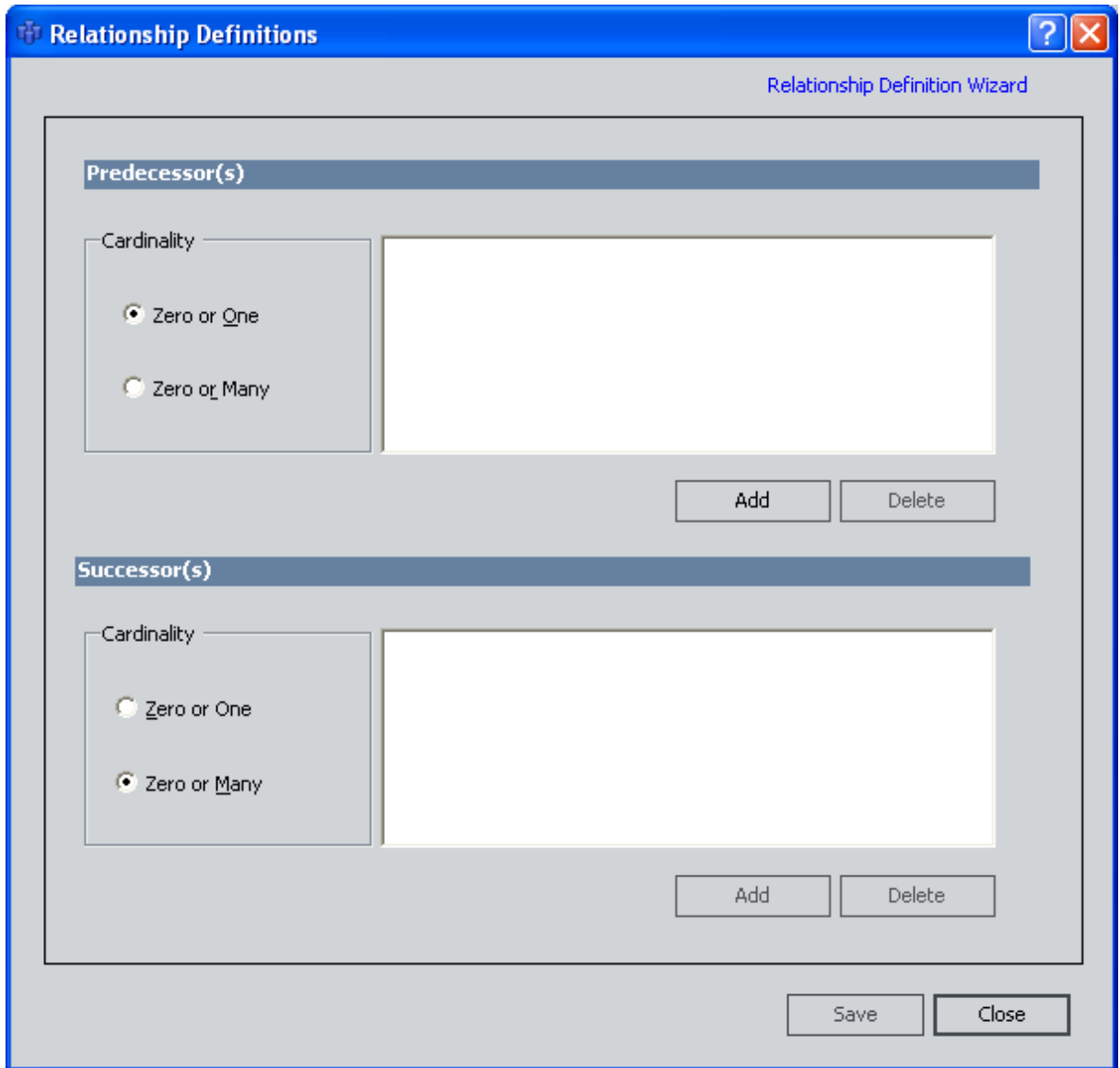
About Creating Relationship Definitions

By creating relationship definitions, you define for a relationship family which entity families are related to one another through that relationship. The relationship definitions that you create between entity families determine which records can be linked to one another. For example, if you create a relationship definition that relates the Pump family to the Work Order family, then Pump records can be linked to Work Order records.

Each relationship definition consists of the following elements:

- A relationship family.

  > 📄 **Note:** The relationship family stores relationship definitions. For example, the Has Maintenance relationship family might store a relationship definition that relates the Pump family to the Work Order family.

- A predecessor entity family.
- A successor entity family.
- Cardinality, which specifies how many entities in the predecessor family can be linked to how many entities in the successor family.

While defining a relationship definition, keep in mind that they do not spread to subfamilies. In other words, you must create relationship definitions for *each* specific predecessor and successor family that you want to participate in the relationship. For example, if the Manufacturer family has two subfamilies, Chinese Manufacturer and Canadian Manufacturer, and you want to be able to link Chinese Manufacturer records and Canadian Manufacturer records to Equipment records, you will need to create a relationship definition for *both* the Chinese Manufacturer family and the Canadian Manufacturer family. Defining a relationship for the Manufacturer family alone would not be sufficient.

After you create relationship definitions, you will need to make sure that you assign family-level privileges such that users have permission to access ALL the families involved in the relationship definition, including the relationship family. Granting permission to only some of the families in a relationship will grant a user only partial access to the records that are linked to one another through that relationship.

> 📄 **Note:** You may want to avoid using inactive entity families in relationship definitions. Otherwise, when viewing a record in the Record Manager, and using the **Show all possible families** option, users will see ALL related families, including inactive families.

Via the Configuration Manager, you can create relationship definitions in two ways:

- By using the **Relationship Definition Wizard** to [define multiple relationships](#).

  -or-

- [Manually](#) via the **Relationship Definitions** window.

⚠ **Note:** If you need to create a relationship definition between two new entity families, you will need to log out of the Configuration Manager application after creating the new families and then log back in before creating the relationship definition. If you attempt to create the relationship definition before logging out of the Configuration Manager application, an error message appears, indicating "There are predecessors without successors."

Creating a Relationship Definition Manually

To create a relationship family manually:

1. In the Configuration Manager, access the **Relationship Definitions** window for the desired family.

2. In the **Predecessor(s)** section, click the **Add** button.

   The **Predecessor Family List** dialog box appears, displaying a list of all predecessor families.

3. Select the desired family from the list, and click **OK**.

4. On the **Relationship Definitions** window, in the **Predecessor(s)** section, select the desired cardinality: **Zero or One** or **Zero or Many**.

5. In the **Successor(s)** section, click the **Add** button.

   The **Successor Family List** dialog box appears, displaying a list of all successor families.

6. Select a family from the list, and click **OK**.

7. On the **Relationship Definitions** window, in the **Successor(s)** section, select the desired cardinality: **Zero or One** or **Zero or Many**.

8. Click the **Save** button.

   The relationship definition that you have created is saved.

Using the Relationship Definition Wizard

Using the **Relationship Definition Wizard**, you can easily create multiple relationship definitions in a single operation.

To create all possible relationships between specified families:

1. In the Configuration Manager, access the **Relationship Definitions** window for the desired family.

2. In the upper, right corner of the window, click the **Relationship Definition Wizard** link.

    The **Relationship Definition Wizard** appears.

3. In the **Predecessor(s)** list, select check boxes next to the desired predecessor families.

    > Note: If you select only one family, you can choose to include all the sub-families by selecting the **Apply to Predecessor sub-families**. If you choose more than one family, the **Apply to Predecessor sub-families**becomes disabled, but you can select subfamilies individually.

4. In the **Successor(s)** list, select the check boxes next to the desired successor families.

    > Note: If you select only one family, you can choose to include all the sub-families by selecting the **Apply to Successor sub-families**. If you choose more than one family, the **Apply to Successor sub-families** becomes disabled, but you can select subfamilies individually.

5. Click the **Next** link.

    The **Relationship Definition Wizard** displays the next step.

6. In the **Predecessor to Successor Cardinality** area, select the desired cardinality: **Zero or One** or **Zero or Many**.

7. In the **Successor to Predecessor Cardinality** area, select the desired cardinality: **Zero or One** or **Zero or Many**.

8. Click the **Finish** link.

    The relationship definitions are created. During the creation process, if the Meridium APM system detects that the relationship already exists between one or more of the families you selected, a message will appear specifying the families for which the relationship has already been defined. You can click the **OK** button to close the message and continue receiving messages about the relationship definitions that already exist, or you can click the **OK to All** button if you do not want the Meridium APM system to display any additional messages about existing relationships that are found.

When all the relationship definitions have been created, the **Relationship Definition Wizard** will close automatically.

Deleting a Predecessor Family

To delete a family from the list of predecessors:

1. In the Configuration Manager, in the **Entity Family/Relationship Family** pane, click the **Relationship Family** tab.

2. Select the desired relationship family from the drop-down box. You can also navigate through the alphabetical list of names from the **Root** folder.

3. In the **Tasks** section, click the **Manage Relationship Definitions** link.

   The **Relationship Definitions** window appears.

4. In the **Predecessor(s)** section, select the predecessor that you want to delete.

5. Click the **Delete** button.

   A confirmation message appears, asking if you really want to delete the predecessor.

6. Click the **Yes** button to confirm the deletion.

   The predecessor family and the related successor family are deleted from the relationship.

Deleting a Successor Family

To delete a family from the list of successors:

1. In the Configuration Manager, in the **Entity Family/Relationship Family** pane, click the **Relationship Family** tab.

2. Select the desired relationship family from the drop-down box. You can also navigate through the alphabetical list of names from the **Root** folder.

3. In the **Tasks** section, click the **Manage Relationship Definitions** link.

   The **Relationship Definitions** window appears.

4. In the **Successor(s)** section, select the successor that you want to delete.

5. Click the **Delete** button.

   A confirmation message appears, asking if you really want to delete the successor.

6. Click the **Yes** button to confirm the deletion.

   The successor family is deleted from the relationship.

Types of Datasheets

Meridium APM offers the following types of datasheets:

- **Standard datasheets:** Datasheets that display fields in a grid. The following image shows a standard datasheet that is configured by default for the Recommendation family.

General Information	Alert		
		Value(s)	
▶	Recommendation ID	REC-1	
	Recommendation Type	General (GENERAL)	
	Originating Reference		
	Recommendation Headline		
	Recommendation Description		...
	Recommendation Basis		
	Asset ID		
	Function ID		
	Required Equipment Status		
	Business Impact		
	Target Completion Date		
	Mandatory Date		
	Recommendation Priority		
	Status	Created (CREATED)	
	Author Name	Duncan, Amy | aduncan	
	Reviewer Name		
	Final Approver Name		
	Assigned To Name		
	Completion Comments		...
	Author Lock	☐	
	Final State Lock	☐	
	Work Request Reference		
	Work Order Number		
	Creation Date	08-Sep-2009	
	Completed Date		
	Published Flag	☐	

> **Note:** The Meridium APM Web Framework supports only standard datasheets. If you want to view the datasheet in the Meridium APM Web Framework, you will need to create a standard datasheet for the desired family.

- **Custom forms:** Datasheets that can be customized to display fields in a more advanced form. Meridium APM provides the following types of custom forms, which you can apply to any family:
    - **[Master/detail](#):** A form that lets you display a record and all the records that are linked to it through a given [relationship definition](#). The following image shows a master/detail datasheet that is configured by default for the Calibration Template family.

- **Custom-layout:** A form that users can customize via the Meridium APM Framework application. Unlike standard datasheets, custom-layout data-sheets provide more flexibility in the layout of fields. For example, they can contain grouped fields, fields that appear side-by-side, and so on. The following image shows a custom form that is configured by default for the Maintenance Item family.

Custom-layout datasheets can be created for any entity family via the Configuration Manager. As part of this creation process, a Security Group must be associated with the custom-layout datasheet. Only Security Users that belong to this Security Group can configure the datasheet in the Meridium APM Framework.

After a custom-layout datasheet has been created in the Configuration Manager, when you create a new record or modify an existing record in the associated family, the datasheet that appears will contain ALL fields that are defined for that family. A Security User that belongs to the associated Security Group can begin customizing the datasheet layout as desired. Note that until the datasheet is customized, the fields will appear in order according to their field sequence order, which is specified via the Configuration Manager application.

If you are familiar with writing code in Visual Studio, you can also create your own custom forms.

Accessing the Datasheet Builder Window

To access the Datasheet Builder window:

1. In the Configuration Manager, in the **Entity Family/Relationship Family** pane, click the **Entity Family** tab.

2. Select the family whose datasheet(s) you want to manage.

3. In the **Tasks** section, click the **Manage Datasheets** link.

   The **Datasheet Builder** window appears, displaying any datasheets that have already been created for the selected family.
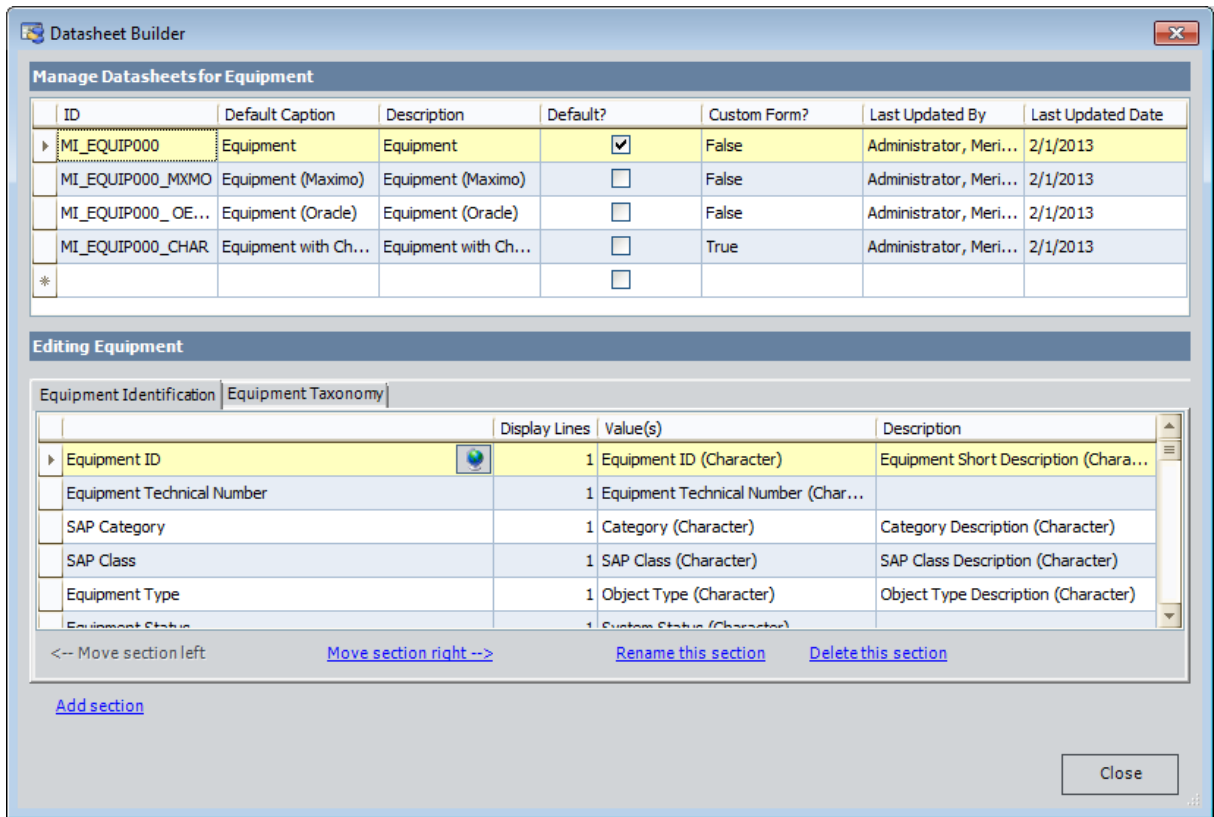
Aspects of the Datasheet Builder Window

The **Datasheet Builder** window lets you view and manage all the datasheets that have been defined for a family. The **Datasheet Builder** window is divided into two main sections:

- A grid at the top, labeled **ManageDatasheets for <Family ID>**, which lists the datasheets that currently exist for the selected family. Each row represents one datasheet.
- A grid at the bottom, labeled **Editing <Datasheet ID>**, which displays the sections and fields that are defined for the datasheet that is selected in the **ManageDatasheets for <Family ID>** grid. Each section appears on a separate tab. Each tab displays the fields defined for that section.

At the bottom of the **Editing <Datasheet ID>** grid, the following links appear:

- **Move section left:** Moves the currently selected section to the left. This link is enabled only when the currently selected section is not the furthest section to the left.
- **Move section right:** Moves the selected section to the right. This link is enabled only when the currently selected section is not the furthest section to the right.
- **Rename this section:** Displays the **Rename Section** dialog box, where you can rename the selected section.
- **Delete this section:** After asking for confirmation, deletes the selected section.
- **Add Section:** Lets you add a new section to the datasheet that is currently selected.

The **Datasheet Builder** window displays one button: **Close**. Clicking this button will close the **Datasheet Builder** window and save any changes that you made to the datasheet. To discard any changes that you made to the datasheet(s), you can close thewindow by clicking the **X** button in the upper, right corner of the window.

About Creating Datasheets

All datasheets are *created* in the Configuration Manager. Standard datasheets are also configured and modified via the Configuration Manager. Custom-layout datasheets, however, can be configured via the Meridium APM Framework application only.

To create a datasheet, you will need to:

- Define the datasheet ID, caption, description, and default status.
- [Add sections to the datasheet](#).
- [Add fields to each section of the datasheet](#).

When creating a datasheet, it is important to organize the sections and fields in a way that will facilitate the data-entry process for the Meridium APM Framework user and Meridium APM Web Framework user.

> 📄 **Note:**The Meridium APM Web Framework does not support custom datasheets. If you want to view a datasheet in the Meridium APM Web Framework, you will need to create a standard datasheet.

Creating a Standard Datasheet

To create a standard datasheet:

1. In the Configuration Manager, access the **Datasheet Builder** window for the family for which you want to create a datasheet.

2. On the **Datasheet Builder** window, in the **Manage Datasheets for <Family ID>** section, click any cell in the first blank row. The row selector will display an image of a pencil, indicating that you can enter information into the row.

3. In the **ID** column, type a unique ID for the datasheet. This ID is required.

4. In the **DefaultCaption** column, type the name that will appear on the datasheet in the Meridium APM Framework application or the Meridium APM Web Framework. The caption is required. Note that you can manage translations for the caption by clicking the icon.

5. If desired, in the **Description** column, type a brief description of the datasheet.

6. If you want this datasheet to be displayed by default when users view records for this family in the Meridium APM Framework application, select the check box in the **Default** column.

7. In the **Custom Form** column, select **False**.

   > ⓘ **Note:** The **Last Updated By** and **Last Updated Date** cells are populated automatically by Meridium APM. These cells display the user ID of the Security User who last modified the datasheet and the date on which those changes were made.

8. Press the Tab key to create a new row.

   > ⓘ **Note:** This step will save the datasheet that you just added and create a new row where you can add an additional datasheet, if desired.

The new datasheet is saved to the database. If you created a standard datasheeet, you can add begin adding sections to the datasheet. If you created a custom form, you can click the **Close** button.

Creating a Master/Detail Datasheet

Before you create a master/detail datasheet, you must first decide:

- Which family will serve as the master family.
- Which family will serve as the detail family.
- Which relationship family will be used to connect them.

Next, you will need to create:

- A datasheet for the detail family.
- A relationship definition that relates the master family to the detail family.

After these prerequisites have been satisfied, you can create the master/detail datasheet. When you do so, you will need to provide the following information:

- The ID of the datasheet that will be used for the master family.

    > Note: The master family datasheet can be a custom form itself.

- The family ID of the detail family.
- The datasheet ID of the datasheet that will be used for the detail family. This will determine which fields appear for the records in the detail grid.

    > Note: The detail family datasheet *cannot* be a custom form.

- The ID of the relationship family that relates the master family to the detail family.

You will want to gather this information before you begin this procedure because you will not have access to the detail family and relationship family information after you begin this process.

These instructions provide details on creating a master/detail datasheet, with the assumption that you are familiar with creating and managing standard datasheets.

To create a master/detail datasheet:

1. In the Configuration Manager, access the **Datasheet Builder** for that family.

2. Create a new datasheet for the master family. For the new datasheet, specify an ID and a default caption. Do not, however, specify any sections or fields for the new datasheet.

3. In the row containing the new datasheet, click the **Custom Form?** cell, and select **True**.

    The **Select Custom Form Assembly** dialog box appears.
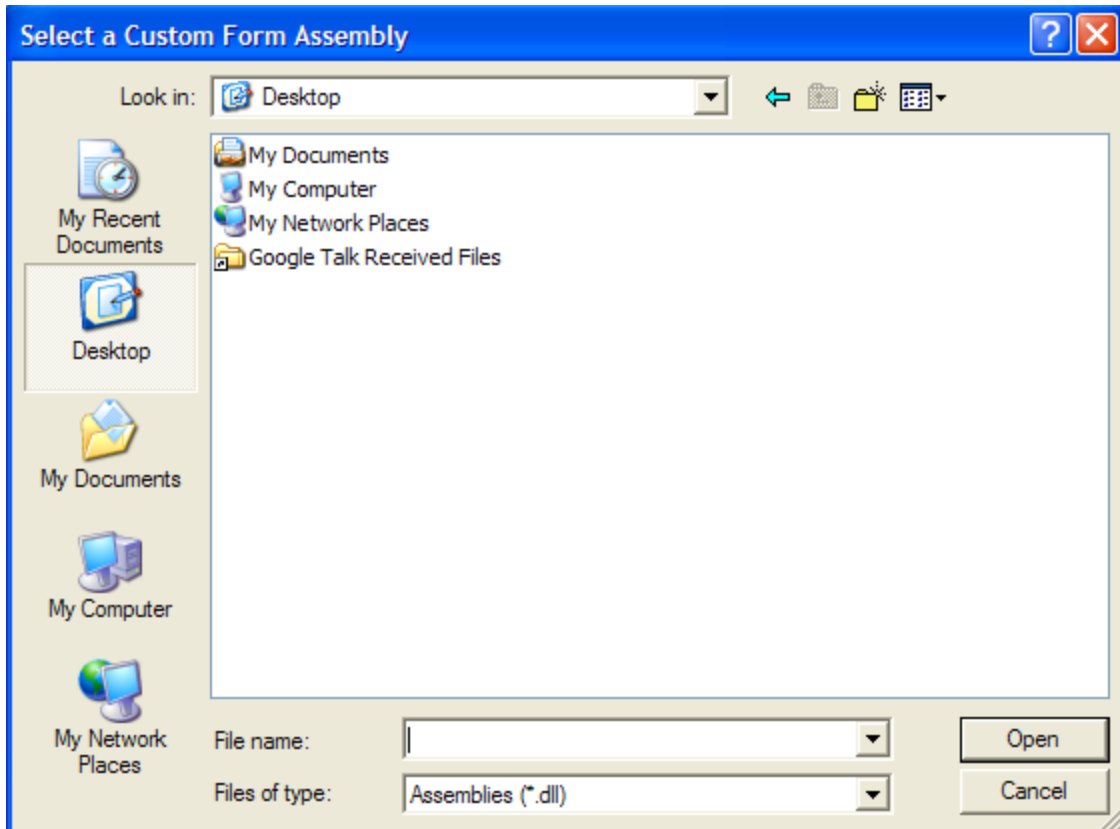
4. Navigate to the folder **<root>\Client\<Versioned Folder>**, where **<root>** is the root drive and folder into which the Meridium APM Framework application has been installed and **<Versioned Folder>** is the version-specific folder for your current version of Meridium APM (e.g., C:\Program Files\Meridium\Client\3.4.0.123.0).

5. Select the file **Meridium.Registry.dll**, and click the **Open** button.

   The **Choose Custom Form** dialog box appears.
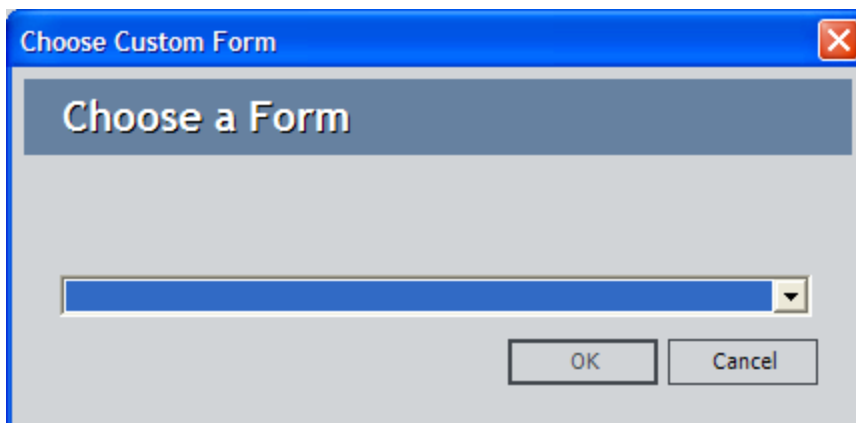
6. In the list, select **Master/Detail Form**, and click **OK**.

The **Custom Form Parameters** dialog box appears, displaying a grid with two columns: **Parameter** and **Value**.
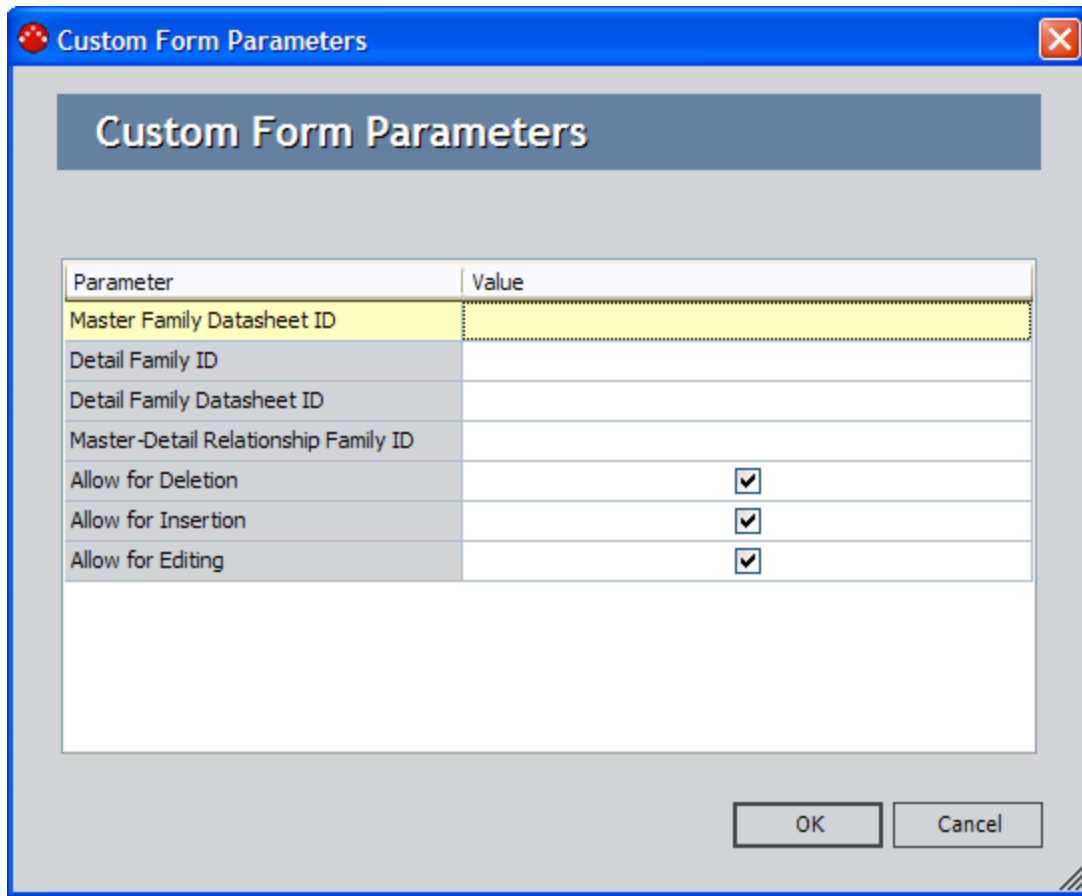
For each item in the **Parameter** column, you will need to specify a value in the **Value** column. The following steps provide details on defining appropriate values.

7. For the **Master Family Datasheet ID** parameter, type the datasheet ID of the datasheet that you want to use for the master family. For this example, enter the ID of the master family datasheet that you created in step 2 above.

8. For the **Detail Family ID** parameter, type the ID of the family that will serve as the detail family.

9. For the **Detail Family Datasheet ID** parameter, type the datasheet ID of the datasheet that will be used for the detail family.

> Note: You must specify a standard datasheet for the Detail Family Datasheet ID parameter. You cannot specify a custom form datasheet.

10. For the **Master-Detail Relationship Family ID** parameter, type the family ID of the relationship family that relates the master family to the detail family.

11. For the **Allow for Deletion** parameter, clear the check box if you do not want users to be able to delete detail records via the master/detail datasheet. By default, this check box is selected, meaning that users *will* be able to delete detail records via the master/detail datasheet.

12. For the **Allow for Insertion** parameter, clear the check box if you do not want users to be able to create new detail records via the master/detail datasheet. By default, this check box is selected, meaning that users *will* be able to create detail records via the master/detail datasheet.

13. For the **Allow for Editing** parameter, clear the check box if you do not want users to be able to modify detail records via the master/detail datasheet. By default, this check box is selected, meaning that users *will* be able to modify detail records via the master/detail datasheet.

> 🗎 **Note:** [These privileges work in conjunction with other security privileges](#). Clearing these check boxes will *revoke* privileges from users who otherwise would be allowed to perform these tasks. These options do not, however, enable permissions for users who do not have the necessary family-level privileges on the detail family.

14. When you have finished entering values for all the parameters, click **OK**.

    The **Datasheet Builder** window appears.

15. Click the **Close** button.

    The master/detail datasheet is created.

Creating a Custom-Layout Datasheet

To create a custom-layout datasheet:

1. In the Configuration Manager, access the **Datasheet Builder** for the desired family.

2. In the first blank row, type an ID and a default caption for the new datasheet.

3. If desired, in the same row, type a description of the datasheet.

4. If you want this datasheet to be the default datasheet for all records in the selected family, select the **Default** check box.

5. In the same row, click the **Custom Form?** cell, and select **True**.

   The **Select Custom Form Assembly** dialog box appears.
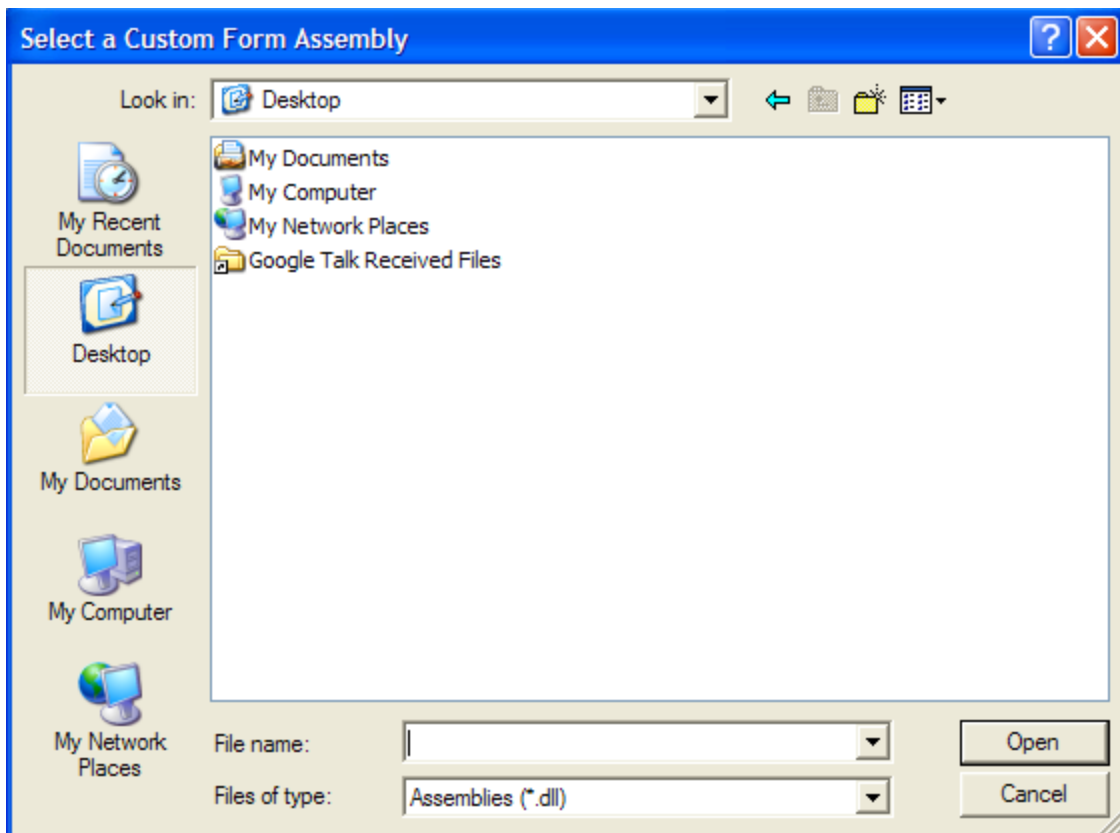
6. Navigate to the folder **<root>\Client\<Versioned Folder>**, where **<root>** is the root drive and folder into which the Meridium APM Framework application has been installed and **<Versioned Folder>** is the version-specific folder for your current version of Meridium APM (e.g., C:\Program Files\Meridium\Client\3.4.0.123.0).

7. Select the file **Meridium.Registry.dll**, and click the **Open** button.

   The **Choose Custom Form** dialog box appears.
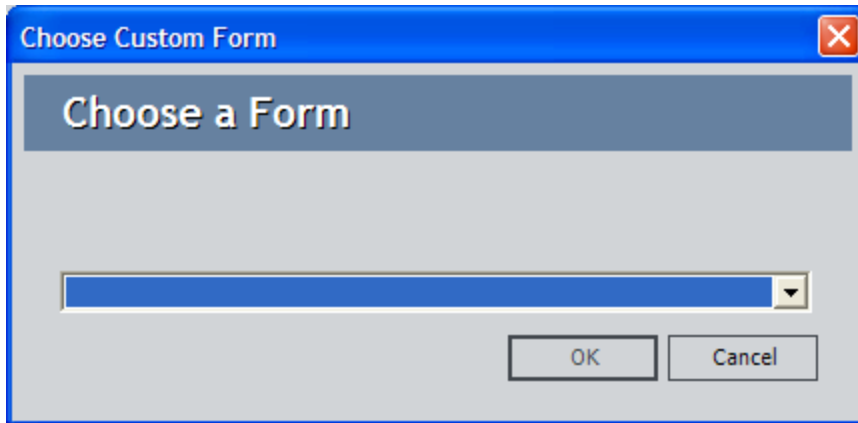
8. In the list, select **Custom-Layout Datasheet**, and click **OK**.

   The **Custom Form Parameters** dialog box appears.
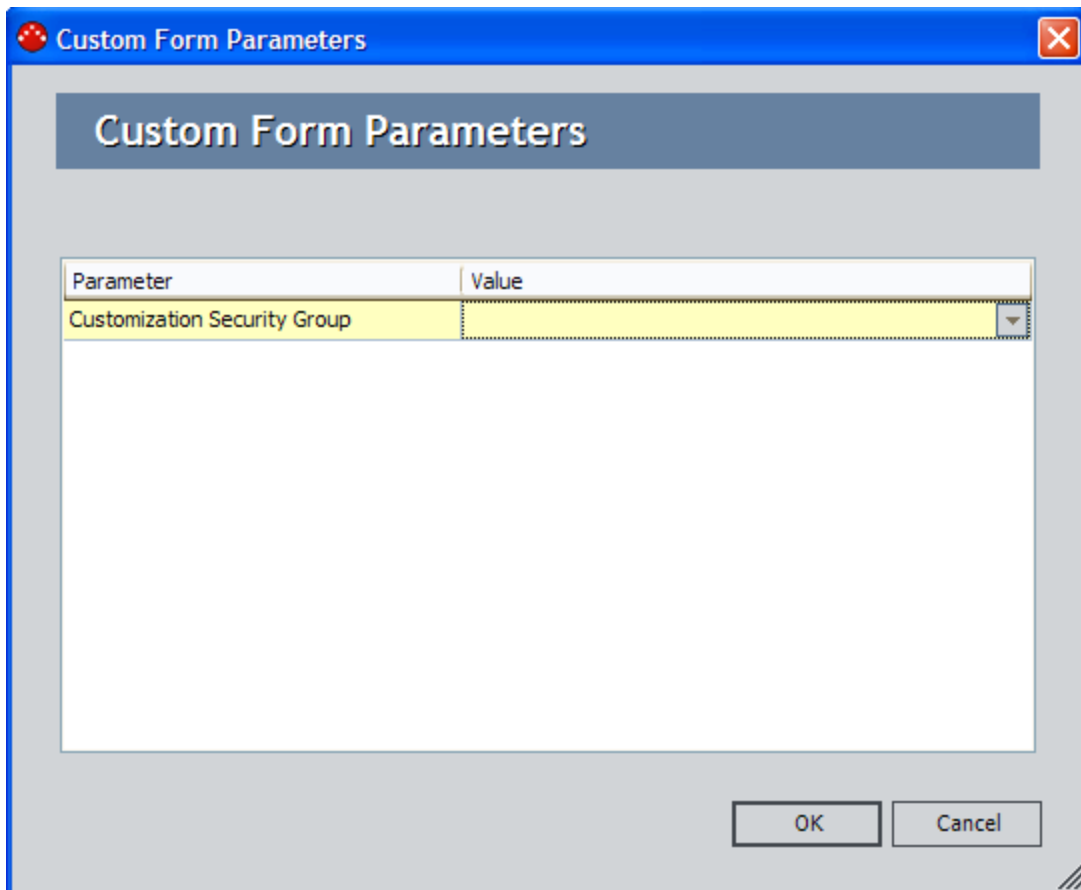
9. In the **Value** list, select the Security Group whose members should be able to modify the layout of the custom-layout datasheet via the Meridium APM Framework application.

10. Click **OK**.

The **Custom Form Parameters** dialog box closes, and the **Datasheet Builder** window returns to focus.

11. Click the **Close** button.

The custom-layout datasheet is created. Security Users who are members of the selected Security Group will now be able to log in to the Meridium APM Framework application and modify the layout of the new datasheet. For details on modifying a custom-layout datasheet via the Meridium APM Framework application, see the Help system in the Meridium APM Framework application.

Designating a Datasheet as the Default Datasheet

The default datasheet is the datasheet that is used by default when a user opens a record belonging to a given family in the Record Manager in the Meridium APM Framework application. A user can select a datasheet other than the default datasheet after the record has been opened. In addition, you can specify via a URL to use a datasheet other than the default datasheet. Because the default datasheet is what will be used unless a different datasheet is selected, you designate as the default datasheet the one that will be most meaningful to most users.

You can designate a datasheet as the default datasheet when you create the datasheet. After a datasheet has been designated as the default datasheet, you can change the selection if you want to designate a datasheet as the default datasheet.

To designate a datasheet as the default datasheet:

1. In the Configuration Manager, access the **Datasheet Builder** window for the family that contains a datasheet that you want to set as the default datasheet for that family.

2. In the **Manage Datasheets for <Family ID>** section, in the row containing the datasheet that you want to set as the default datasheet, select the **Default** check box.

   > ⓘ **Note:** If a datasheet is already set as the default datasheet, selecting the **Default** check box for another datasheet will result in the *temporary* selection of *both* datasheets as the default datasheet. When you move your cursor to another row or close the **Datasheet Builder** window, the last datasheet that you set as the default datasheet will remain the default datasheet, and the **Default** check box in the other row will be cleared automatically.

3. Click the **Close** button, or move your cursor to another row.

   Your changes are saved to the database.

Modifying Existing Datasheets

To modify an existing datasheet:

1. In the Configuration Manager, <u>access the **Datasheet Builder** window</u> for the family containing the datasheet that you want to modify.

2. In the **Manage Datasheets for <Family ID>** section, which appears at the top of the screen, modify the ID, Default Caption, Description, Default setting, or Custom Form designation for any datasheet in the list.

3. When you are finished modifying the datasheet, click the **Close** button.

   The **Datasheet Builder** window closes, and your changes are saved to the database.

About Field Background Colors in Datasheets

Via the Configuration Manager, you can define rules for fields that will determine how they behave when users interact with them in datasheets. Fields with certain types of rules can be highlighted with color-coding to indicate to users how they work. For instance, the following image shows a required field whose background is colored pink.
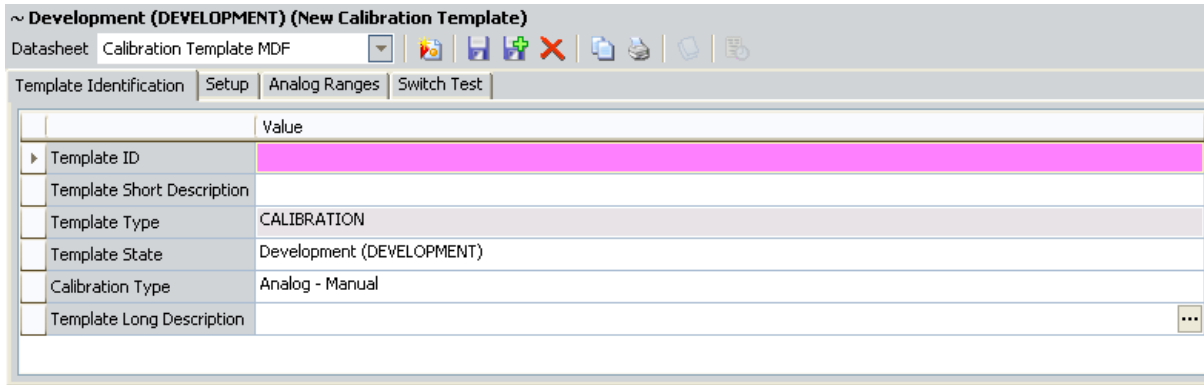
Each of type of field that can be color-coded has a default background color. The following list contains the field types that can be color-coded and the default background color for each:

- **Required fields:** Red
- **Disabled fields:** Gray
- **Multi-value fields:** White
- **Formula fields:** White

For any of these types of fields, you can modify the default color and use a custom background color instead. For details on modifying the default background colors, click any of the following links:

- [Defining the Background Color for Required Fields](#)
- [Defining the Background Color for Disabled Fields](#)
- [Defining the Background Color for Multi-Value Fields](#)
- [Defining the Background Color for Formula Fields](#)

> **Note:** Field background colors will not be used on the MI Human Resource datasheet that is defined for the Human Resource family (e.g., required fields will still be colored red, regardless of the custom background color that you define for required fields).

Defining the Background Color for Required Fields

To define the color of required fields:

1. In the Configuration Manager, on the main menu, click **Tools**, point to **Global Preferences**, and then click **Required Field BackColor**.

   The **Color** dialog box appears.

2. In the list of **Basic Colors**, click the color that you want to use.

   -or-

   Click the **Define Custom Colors** button to define a custom color.

3. Click **OK**.

   Your color selection is saved.

Defining the Background Color for Disabled Fields

To define the color of Disabled fields:

1. In the Configuration Manager, on the main menu, click **Tools**, point to **Global Preferences**, and then click **Disabled Field BackColor**.

    The **Color** dialog box appears.

2. In the list of **Basic Colors**, click the color that you want to use.

    -or-

    Click the **Define Custom Colors** button to define a custom color.

3. Click **OK**.

    Your color selection is saved.

Defining the Background Color for Multi-Value Fields

To define the color of Multi-Value fields:

1. In the Configuration Manager, on the main menu, click **Tools**, point to **Global Preferences**, and then click **Multi-Value Field BackColor**.

   The **Color** dialog box appears.

2. In the list of **Basic Colors**, click the color that you want to use.

   -or-

   Click the **Define Custom Colors** button to define a custom color.

3. Click **OK**.

   Your color selection is saved.

Defining the Background Color for Formula Fields

To define the color of Formula fields:

1. In the Configuration Manager, on the main menu, click **Tools**, point to **Global Preferences**, and then click **Formula Field BackColor**.

   The **Color** dialog box appears.

2. In the list of **Basic Colors**, click the color that you want to use.

   -or-

   Click the **Define Custom Colors** button to define a custom color.

3. Click **OK**.

   Your color selection is saved.

•

Adding a Section to a Datasheet

Datasheet sections help organize fields in a logical manner. Each section can contain fields that are related to one another. If you are configuring a datasheet for a family that contains a large number of fields, you will likely want to create multiple sections for the datasheet. Each section will appear on a separate tab in the **Datasheet Builder** window in the Configuration Manager and in the Record Manager in the Meridium APM Framework application and the Meridium APM Web Framework.

> ⓘ **Note:** Before you can add a section to a datasheet, you must define the ID and Default Caption for the datasheet.

To add a section to a datasheet:

1. In the Configuration Manager, on the **Datasheet Builder window**, below the **Editing <Datasheet ID>** section, click the **Add Section** link.

   The **New Section Name** dialog box appears.

2. In the **Section Name** text box, type a name for the section you are creating. This name will appear on the section tab for this section. You can manage translations for the section name by clicking the 🌐 button.

3. Click **OK**.

   The **New Section Name** dialog box closes, returning focus to the **Datasheet Builder** window. The new section appears in the **Editing <Datasheet ID>** section of the window.

After you have created a new section, you can choose which fields will appear in that section of the datasheet.

Adding Fields to a Section

Because datasheets are family specific, each datasheet can contain any field that has been defined for that family. By placing fields on a datasheet, you are specifying the location in the physical table that data will be stored when users enter values into those fields. Fields are assigned to sections, so before you can add fields to a datasheet, you must create at least one section.

The datasheet is not required to contain all the fields that are defined for a family. Each datasheet, however, can use each family field only *once*. If more than one datasheet has been defined for a family, each datasheet can use each field one time.

> Note: Binary fields should not be added to datasheets. If a binary field exists on a datasheet, when a user enters a value in the field and tries to save the record, an error message will appear, indicating that the record cannot be saved.

To add fields to a datasheet section:

1. In the Configuration Manager, on the **Datasheet Builder** window, in the first blank row in the desired **Editing <Datasheet ID>** section, click the cell in the **Value(s)** column.

2. In the list, select the desired field. The list contains all the active fields that are defined for the family that have not been added to the current datasheet. Inactive fields do not appear in the list.

   > Hint: If no list appears in the cell, it means that all available fields have already been added to the datasheet.

3. In the same row, click the cell to the left of the **Display Lines** cell. The field is populated with the field caption of the field that you selected in the **Value(s)** cell. You can modify this value if desired. Note that the label that appears in the datasheet is not required to match the field caption, but the datasheet caption cannot exceed 100 characters in length. Note that you can manage translations for the field label by clicking the ⊕ button.

   > Note: Defining localized datasheet captions alone will not ensure that localized values appear everywhere throughout the Meridium APM Framework application. If you want to use localized values, you should define them both for field captions and for datasheet captions.

4. In the same row, click the **Display Lines** cell. This cell specifies how many lines will be used to display data in this field when the datasheet is displayed in the Record Manager. The default value is **4** (four) for text fields and **1** (one) for all other fields, but you can select any value up to 15 for any field.

5. Continue adding fields as desired.

6. When you have finished adding fields to the section, click the **Close** button.

Your changes are saved.

About Modifying Datasheet Sections

After you have added one or more sections to a datasheet, you can modify a section by:

- [Moving a section (tab) to the left or the right](#) using the **Move section left** or **Move section right** link.
- [Modifying the datasheet caption](#) for any field that exists in a section.
- [Rearrange the order in which fields will appear on the datasheet](#).

> ⓘ **Note:** You cannot move rows into the last, empty row, which is meant to be used for adding new fields to the datasheet. Doing so will cause an error message to appear.

- [Renaming the section](#).
- [Removing a field from the section](#).

•

Rearranging Sections

After a datasheet contains more than one section, you can move those sections to the left or the right. The order of the sections in the **Datasheet Builder** window determines the order of the tabs in the Meridium APM Framework application and the Meridium APM Web Framework.

To rearrange datasheet sections:

1. In the Configuration Manager, on the **Datasheet Builder window**, in the **ManageDatasheets for <Family ID>** grid, select the datasheet that contains the section that you want to move to the left or the right.

2. In the **Editing [Datasheet ID]** grid, select the section that you want to move.

3. To move the section to the left, click the **Move section left** link. Note that this link is enabled only when the currently selected section is not the furthest section to the left.

   -or-

   To move the section to the right, click the **Move section right** link. Note that this link is enabled only when the currently selected section is not the furthest section to the right.

4. When the sections are in the desired order, click the **Close** button.

   The **Datasheet Builder** window closes, and your changes are saved.

Modifying the Datasheet Caption for a Field

To modify the datasheet caption for a field:

1. In the Configuration Manager, on the **Datasheet Builder** window, in the **ManageDatasheets for <Family ID>** grid, select the datasheet that contains the field whose datasheet caption you want to modify.

2. In the **Editing [Datasheet ID]** grid, select the section that contains the field whose datasheet caption you want to modify.

3. In the selected section, select the row that contains the field whose datasheet caption you want to modify.

4. In the selected row, in the cell to the left of the **Display Lines** cell, type the desired datasheet caption.

5. Click the **Close** button.

   The **Datasheet Builder** window closes, and the changes are saved.

Rearranging Rows

To change the order of rows in a datasheet:

1. In the Configuration Manager, on the **Datasheet Builder** window, in the **ManageDatasheets for <Family ID>** grid, select the datasheet that contains the fields that you want to rearrange.

2. In the **Editing <Datasheet ID>** grid, select the section that contains the rows that you want to rearrange.

3. In the selected section, click the left mouse button to select the row that contains the field that you want to move.

4. Drag the row up or down to the desired location.

   > 🗋 **Note:** Do not drag the row to the bottom of the grid (i.e., the last empty row). Doing so will cause an error message to be displayed. If you want to move a field to the *bottom* of a datasheet section, you will need to delete the field from the section and then add it back.

5. Release the left mouse button.

   The row is placed *above* the row on which your cursor was placed when you released the mouse button.

6. Continue in this way, moving rows up or down as desired, until all rows are in the desired position.

7. Click the **Close** button.

   The **Datasheet Builder** window closes, and your changes are saved.

Renaming a Section

To rename a section that currently exists for a datasheet:

1. In the Configuration Manager, on the **Datasheet Builder** window, in the **ManageDatasheets for <Family ID>** grid, select the datasheet that contains the section that you want to rename.

2. In the **Editing <Datasheet ID>** grid, select the section that you want to rename.

3. Click the **Rename this section** link.

    The **Rename Section** dialog box appears, displaying the name that is currently assigned to this section.
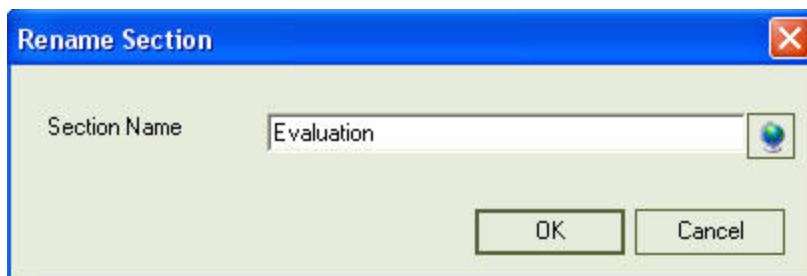
4. In the **Section Name** text box, delete the existing name and type the new name.

5. If desired, click the button to manage translations for that string.

6. Click **OK**.

    The section label on the **Datasheet Builder** window is updated to reflect the new section name.

Removing a Field from a Section

To remove a field from a section in a datasheet:

1. In the Configuration Manager, access the **Datasheet Builder** window for the family to which the datasheet belongs.

2. In the **Manage Datasheets for <Family ID>** section, click the row representing the datasheet to which the field belongs.

3. In the **Editing <Datasheet ID>** section, click the section that contains the field that you want to remove.

4. In the selected section, select the row that contains the field that you want to remove.

5. In the selected row, right-click the **Display Lines** cell or the cell to the right of the **Display Lines** cell, and click **Delete Row**.

   The field is removed the datasheet.

Deleting a Section

To delete a section from a datasheet:

1. In the Configuration Manager, access the **Datasheet Builder** for the family to which the datasheet belongs.

2. In the **Manage Datasheets for <Family ID>** section, click the row of the datasheet to which the section belongs.

3. In the **Editing <Datasheet ID>** section, click the tab of the section you want to delete.

4. Click the **Delete this section** link.

   The section is deleted.

About Master/Detail Datasheets

A *master/detail datasheet* is a custom form that lets you display a record and all the records that are linked to it through a given [relationship definition](#). Master/detail datasheets can be configured for any entity family using any relationship definition that relates it to another entity family.

For example, in the baseline Meridium APM database, the Calibration Template family is related to the Calibration Template Detail family through the Has Template Details relationship. The Meridium APM product includes a master/detail datasheet that allows you to open a Calibration Template record while simultaneously viewing, creating, and modifying linked Calibration Template Detail records.

The following figure shows an example of how this master/detail datasheet might look when you are viewing records in the Record Manager.

Privileges for Master/Detail Datasheets

Like standard datasheets, access to records displayed using master/detail datasheets is controlled by family-level security privileges. But because a master/detail datasheet displays records belonging to *two* entity families that are related to one another through a relationship family, the security considerations for master-detail datasheets are somewhat more complex. Specifically:

- To access records in a master/detail datasheet, at least *View* privileges are required on the master family. Even if users have privileges on the detail family, without privileges on the master family, they will not be able to access records that are displayed in master/detail datasheets.

- To view existing *detail* records, users must have View privileges on the detail family and the relationship family that relates the master family to the detail family.

- To create new *detail* records in a master/detail datasheet, users must have View, Insert, and Update privileges on the detail family.

In addition to family-level privileges, the ability to manage records in master/detail datasheets is controlled by datasheet-level privileges. You can define datasheet-level privileges on the **Custom Form Parameters** dialog box, which appears when you create a master/detail datasheet.

Three datasheet-level permissions options are available and control users' ability to manage *detail* records in a master/detail datasheet:

- **Allow for Deletion:** Determines whether users will be allowed to *delete* detail records.
  - When this check box is selected, users with Delete privileges on the detail family will be allowed to delete detail records.
  - When this check box is cleared, users with Delete privileges on the detail family will not be allowed to delete detail records.
  - Regardless of whether this check box is selected or cleared, users *without* Delete privileges on the detail family will not be allowed to delete detail records.
- **Allow for Insertion:** Determines whether users will be allowed to *create* detail records.
  - When this check box is selected, users with Insert privileges on the detail family will be allowed to create detail records.
  - When this check box is cleared, users with Insert privileges on the detail family will not be allowed to create detail records.
  - Regardless of whether this check box is selected or cleared, users *without* Insert privileges on the detail family will not be allowed to create detail records.
- **Allow for Editing:** Determines whether users will be allowed to *modify* detail records.
  - When this check box is selected, users with Update privileges on the detail family will be allowed to modify detail records.
  - When this check box is cleared, users with Update privileges on the detail family will not be allowed to modify detail records.
  - Regardless of whether this check box is selected or cleared, users *without* Update privileges on the detail family will not be allowed to modify detail records.
- **Allow for Linking:** Determines whether users will be allowed to link new detail records to the master record.
  - When this check box is selected, users will be allowed to link detail records to the master record if they have Insert privileges on the relationship family that relates the detail family to the master family.
  - When this check box is cleared, users will not be allowed to link detail records to the master record even if they have Insert privileges on the relationship family that relates the detail family to the master family.
  - Regardless of whether this check box is selected or cleared, users *without* Insert privileges on the relationship family that relates the detail family to the master family will not be allowed to link detail records to the master record.

- **Allow for Unlinking:** Determines whether users will be allowed to unlink existing detail records from the master record.
    - When this check box is selected, users will be allowed to unlink detail records from the master record if they have Delete privileges on the relationship family that relates the detail family to the master family.
    - When this check box is cleared, users will not be allowed to unlink detail records from the master record even if they have Delete privileges on the relationship family that relates the detail family to the master family.
    - Regardless of whether this check box is selected or cleared, users *without* Delete privileges on the relationship family that relates the detail family to the master family will not be allowed to unlink detail records from the master record.

> **Note:** Family-level privileges alone control a user's ability to manage *master* records.

About Custom-Layout Datasheets

A *custom-layout datasheet* is a form-based datasheet whose layout can be configured via the Meridium APM Framework application. Unlike standard datasheets, custom-layout datasheets provide more flexibility in the layout of fields. For example, they can contain grouped fields, fields that appear side-by-side, and so on. The following image shows a custom-layout datasheet that is configured in the baseline ASI module for the Maintenance Item family.

When you create a custom-layout datasheet, you will need to select a Security Group. Only members of that Security Group will be able to modify the layout of the datasheet via the Meridium APM Framework application.

> **Note:** All users with the appropriate family-level security privileges will be able to modify field values via the datasheet even if they cannot modify the layout of the datasheet itself.

Note that beyond changing the ID, caption, description, default status, and default field order, which is based on the field sequence order, you cannot modify a custom-layout datasheet via the Configuration Manager. The layout of the datasheet must be configured via the Meridium APM Framework application.

Accessing the Customize <Family Caption> Datasheet Window

To customize a custom-layout datasheet, you will need to access the **Customize <Family Caption> Datasheet** window. Note that in order to customize a custom-layout datasheet, you must be a member of the Security Group that is associated with the custom-layout datasheet. A Security Group can be associated with a custom-layout datasheet via the Configuration Manager.

To access the Customize <Family Caption> Datasheet window:

- In the Meridium APM Framework, in any record that uses a custom-layout datasheet, right-click any field caption, and click **Customize Layout**.

  The **Customize <Family Caption> Datasheet** window appears, where **<Family Caption>** is the caption of the current record's family.
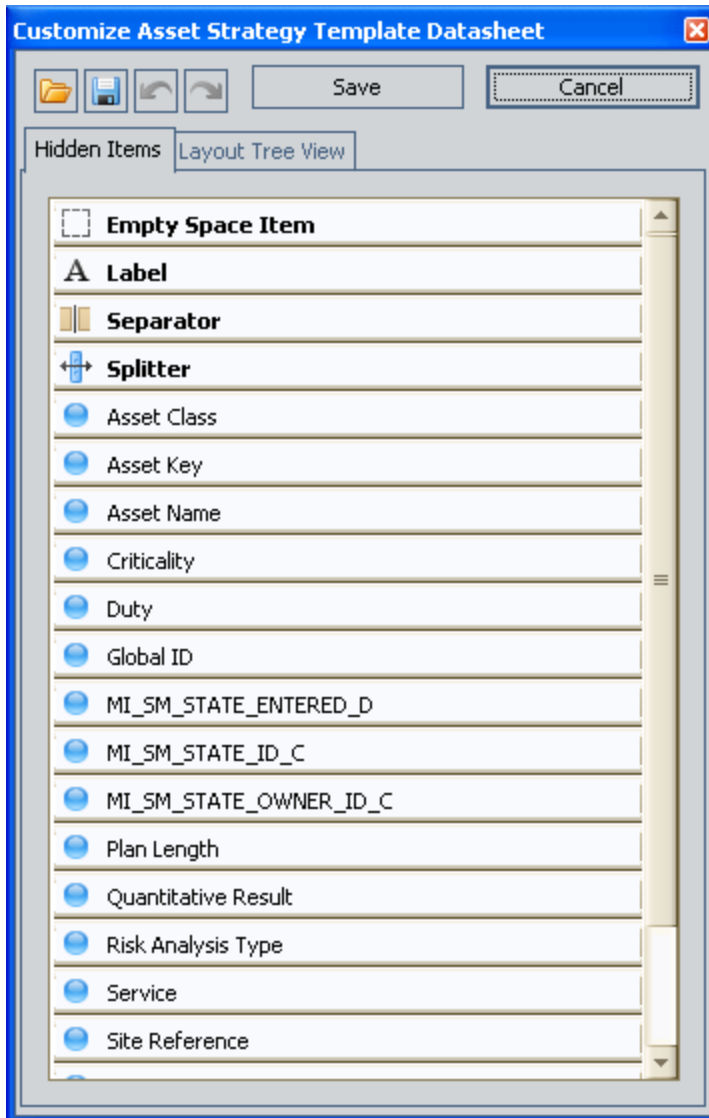
Aspects of the Customize <Family Caption> Datasheet Window
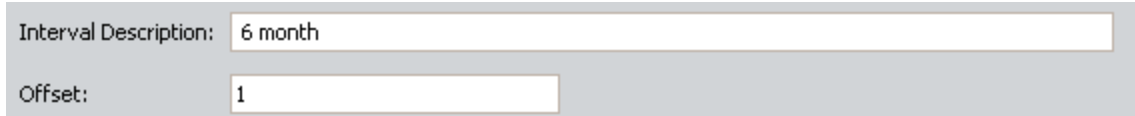
The Customize <Family Caption> Datasheet window contains the following features:

- **Toolbar:** Contains the following buttons:
    - : Displays the **Open** dialog box, where you can select an XML file to use for importing a custom-layout datasheet configuration.
    - : Displays the **Save As** dialog box, where you save the current datasheet configuration as an XML file.
    - : Undoes the last action. Note that you can click this button multiple times to undo multiple actions.
    - : Performs the last action that was undone. Note that if you undid multiple actions in a row, you can redo those actions by clicking this button multiple times.
    - **Save:** Saves any unsaved changes that you have made to the datasheet configuration.
    - **Cancel:** Closes the **Customize <Family Caption> Datasheet** window without saving any changes.
- **Hidden Items tab:** Contains the following features:
    - **Empty Space Item option:** An option that you can drag to the datasheet to create an extra space between datasheet items.
    - **Splitter option:** An option that you can drag to the datasheet to create a split bar between datasheet items.
    - **List of hidden items:** A list of items that are hidden from the datasheet. If desired, you can drag any hidden item to the datasheet to display it.
- **Layout Tree View tab:** Displays a hierarchical view of the items that are displayed in the datasheet.

Types of Items on Custom-Layout Datasheets
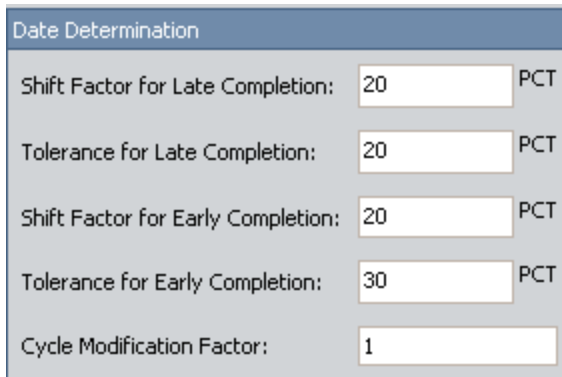
Custom-layout datasheets can contain the following types of items:

- Individual fields that are not grouped under a single heading.
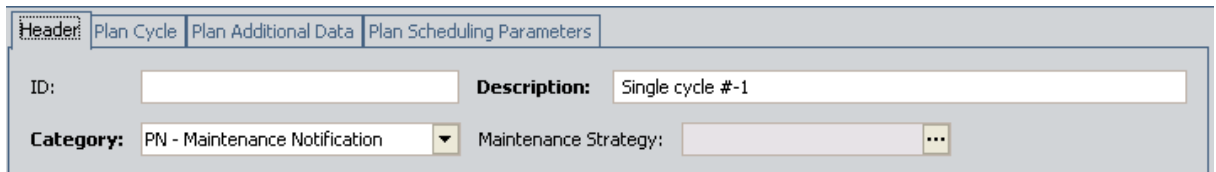
Interval Description:	6 month
Offset:	1

- Groups of fields that are organized under a single heading.

Date Determination

Shift Factor for Late Completion:	20	PCT
Tolerance for Late Completion:	20	PCT
Shift Factor for Early Completion:	20	PCT
Tolerance for Early Completion:	30	PCT
Cycle Modification Factor:	1	

- Tabs.

Header | Plan Cycle | Plan Additional Data | Plan Scheduling Parameters

ID:		Description:	Single cycle #-1
Category:	PN - Maintenance Notification	Maintenance Strategy:	

- Split bars between items.

Interval:	20	Interval Unit:	DAY

- Extra space between items.

Interval:	20	Interval Unit:	DAY
Interval Description:	6 month		

Before a family's custom-layout datasheet is modified, when you open a record in that family and view the custom-layout datasheet, you will see a list of all of the fields that are defined for the family. If you are a member of the Security Group that is associated with the custom-layout datasheet, you can begin modifying the datasheet layout by adding, moving, or renaming items.

Moving Items

To move an item on a custom-layout datasheet:

1. In the Meridium APM Framework, open or create a record that uses a custom-layout datasheet whose item you want to move.

2. Access the **Customize <Family Caption> Datasheet** window.

3. In the datasheet, select the item that you want to move.

4. Drag the item to the desired location. You can drag the item above, below, to the left, or to the right of any other item.

   The item moves to the selected location.

5. On the **Customize <Family Caption> Datasheet** window, click the 💾 button.

   Your changes are saved.

Resizing a Field

To resize a field on a custom-layout datasheet:

1. In the Meridium APM Framework, open or create a record that uses a custom-layout datasheet whose field you want to resize.

2. Access the **Customize <Family Caption> Datasheet** window.

3. In the datasheet, select the field that you want to resize.

4. Drag the field until it reaches the desired size. Note that if you want to increase the height of the field, you must drag the field using the *bottom* of the field selector box. For example, in the following image, the Work Center field is being resized.

   ↕

   The field size changes.

5. On the **Customize <Family Caption> Datasheet** window, click the 🖫 button.

   Your changes are saved.

Hiding Items

You can hide the following items on a custom-layout datasheet:

- Fields
- Groups
- Split bars
- Empty spaces

To hide an item on a custom-layout datasheet:

1. In the Meridium APM Framework, open or create a record that uses a custom-lay-out datasheet containing an item that you want to hide.

2. Access the **Customize <Family Caption> Datasheet** window.

3. In the datasheet, right-click the item that you want to hide, and click **Hide Item**.

   The item is removed from the datasheet and added to the **Hidden Items** tab on the **Customize <Family Caption> Datashee**t window.

   > *ⓘ* **Hint:** If you hid an item by mistake, you can show it again by dragging it from the **Hidden Items** tab to the datasheet.

4. On the **Customize <Family Caption> Datasheet** window, click the 🖫 button.

   Your changes are saved.

Showing Hidden Items

If you have hidden an item on a custom-layout datasheet, you can use the following instructions to show the item again. The following hidden items can be shown again:

- Fields
- Groups
- Split bars
- Empty spaces

To show a hidden item on a custom-layout datasheet:

1. In the Meridium APM Framework, open or create a record that uses a custom-layout datasheet containing a hidden item that you want to show.

2. Access the **Customize <Family Caption> Datasheet** window.

3. On the **Customize <Family Caption> Datasheet** window, on the **Hidden Items** tab, select the item that you want to show.

4. Drag the selected item to the desired location on the datasheet.

   The item appears on the datasheet in the selected location.

5. On the **Customize <Family Caption> Datasheet** window, click the 🖫 button.

   Your changes are saved.

Adding Extra Space Between Items

You can add extra space between items on a custom-layout datasheet to provide separation between items.

To add extra space between items on a custom-layout datasheet:

1. In the Meridium APM Framework, open or create a record that uses a custom-layout datasheet to which you want to add extra space.

2. Access the **Customize <Family Caption> Datasheet** window.

3. On the **Customize <Family Caption> Datasheet** window, if it is not already selected, select the **Hidden Items** tab.

4. On the **Hidden Items** tab, drag the **Empty Space Item** option to the location on the datasheet to which you want to add extra space.

   A bar representing an empty space appears in the selected location. In the following image, an arrow points to the bar that represents the empty space. This arrow is intended to highlight functionality and is not actually part of the datasheet.
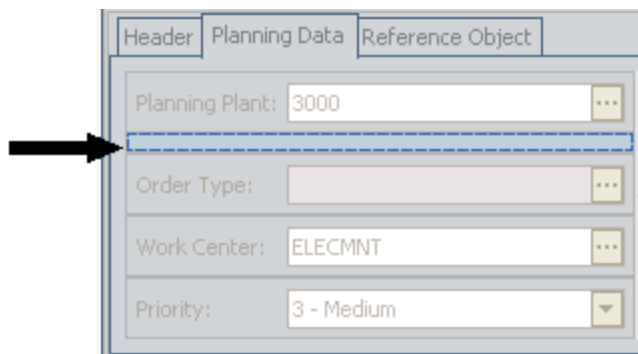
   When a user views the datasheet, the empty space will look like the following image.
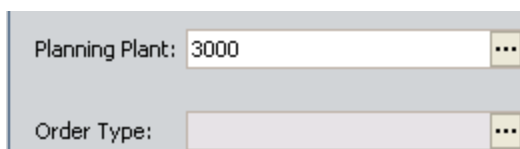
   > ⓘ **Hint:** You can add as many empty spaces to a single area of a datasheet as you prefer. To add more space than you can create with one empty space item, repeat step 4 of these instructions as many times as necessary.

5. On the **Customize <Family Caption> Datasheet** window, click the 💾 button.

   Your changes are saved.

Adding a Label

You can add a label to a custom-layout datasheet to provide details about the datasheet items that are displayed under that label.

To add a label to a custom-layout datasheet:

1. In the Meridium APM Framework, open or create a record that uses a custom-lay-out datasheet to which you want to add a label.

2. Access the **Customize <Family Caption> Datasheet** window.

3. On the **Customize <Family Caption> Datasheet** window, if it is not already selected, select the **Hidden Items** tab.

4. On the **Hidden Items** tab, drag the **Label** option to the location on the datasheet to which you want to add extra space.

   A bar representing a label appears in the selected location. The default label is **LabelItem<n>**, where **<n>** is the number of labels that have been added to this datasheet. In the following image, an arrow points to the bar that represents the label. This arrow is intended to highlight functionality and is not actually part of the datasheet.

   Using the default label, when a user views the datasheet, the label will look like the following image.
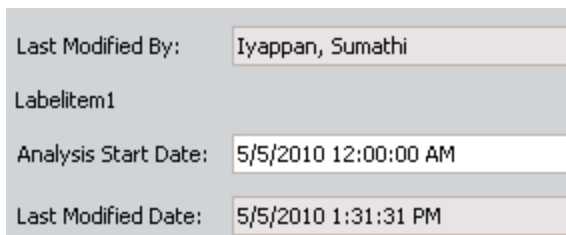
   To make the label useful to users, you will need to rename it.

5. On the **Customize <Family Caption> Datasheet** window, click the button.

   Your changes are saved.

Adding a Separator

You can add a separation line to a custom-layout datasheet to provide visual distinction between the items on either side of the line.

To add a separator to a custom-layout datasheet:

1. In the Meridium APM Framework, open or create a record that uses a custom-lay-out datasheet to which you want to add a separator.

2. Access the **Customize <Family Caption> Datasheet** window.

3. On the **Customize <Family Caption> Datasheet** window, if it is not already selected, select the **Hidden Items** tab.

4. On the **Hidden Items** tab, drag the **Separator** option to the location on the datasheet to which you want to add the separation line.

   A bar representing a separator appears in the selected location. In the following image, an arrow points to the bar that represents the separator. This arrow is intended to highlight functionality and is not actually part of the datasheet.
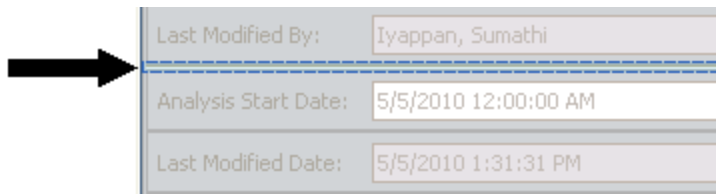
   When a user views the datasheet, the separator will look like the following image.
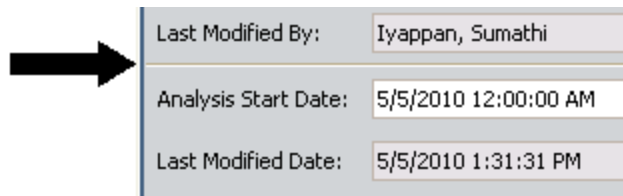
5. On the **Customize <Family Caption> Datasheet** window, click the 💾 button.

   Your changes are saved.

Adding Split Bars Between Items

You can add split bars to a datasheet so that when you drag the split bar, the items on either side of the split bar grow larger or smaller, depending on the direction in which you drag the bar.

To add a split bar between items on a custom-layout datasheet:

1. In the Meridium APM Framework, open or create a record that uses a custom-layout datasheet to which you want to add a split bar.

2. Access the **Customize <Family Caption> Datasheet** window.

3. On the **Customize <Family Caption> Datasheet** window, if it is not already selected, select the **Hidden Items** tab.

4. On the **Hidden Items** tab, drag the **Splitter** option to the location on the datasheet to which you want to add a split bar.

   A split bar appears in the selected location.

5. On the **Customize <Family Caption> Datasheet** window, click the button.

   Your changes are saved.

Controlling an Item's Size

You can impose size constraints to determine how fields and extra spaces on a custom-layout datasheet should be resized when users resize the Meridium APM Framework main window. By default, items will be resized according to the size of the Meridium APM Framework main window so that the proportions always remain consistent. If desired, you can lock an item's size so that it will not be resized vertically, horizontally, or in either direction.

To control the size of a field or extra space when users resize the Meridium APM Framework main window:

1. In the Meridium APM Framework, open or create a record that uses a custom-layout datasheet containing an item whose size you want to control.

2. Access the **Customize <Family Caption> Datasheet** window.

3. On the datasheet, right-click the field or extra space whose size you want to control.

   A shortcut menu appears.

4. On the shortcut menu, point to **Size Constraints**, and select one of the following options:

   - **Reset to default:** Causes the item to be resized according to the size of the Meridium APM Framework main window so that the proportions always remain consistent.

   - **Free sizing:** Causes the item to be resized according to the size of the Meridium APM Framework main window so that the proportions always remain consistent.

   - **Lock Size:** Causes the item's width and height to remain unchanged when the Meridium APM Framework main window is resized.

   - **Lock Width:** Causes the item's width to remain unchanged when the Meridium APM Framework main window is resized.

   - **Lock Height:** Causes the item's height to remain unchanged when the Meridium APM Framework main window is resized.

5. On the **Customize <Family Caption> Datasheet** window, click the 🖫 button.

   Your changes are saved.

Modifying Item Labels

If desired, you can modify field labels, group labels, and tab labels.

To modify a label on a custom-layout datasheet:

1. In the Meridium APM Framework, open or create a record that uses a custom-layout datasheet containing a label that you want to modify.

2. Access the **Customize <Family Caption> Datasheet** window.

3. In the datasheet, right-click the label that you want to modify, and click **Rename**.

   The selected label is highlighted in blue.

4. Type a new label for the item, and press Enter when you are finished.

   The label is modified.

5. On the **Customize <Family Caption> Datasheet** window, click the 🖫 button.

   Your changes are saved.

Hiding Item Labels

If desired, you can hide a field label, group label, or tab label on a custom-layout data-sheet.

> ⓘ **Note:** You cannot hide a label for a logical field.

To hide a label on a custom-layout datasheet:

1. In the Meridium APM Framework, open or create a record that uses a custom-lay-out datasheet containing a label that you want to hide.

2. Access the **Customize <Family Caption> Datasheet** window.

3. In the datasheet, right-click the label that you want to hide, and click **Hide Text**.

   The label is removed from the datasheet.

   > ⓘ **Note:** If you hid the label by mistake, you can click the button on the **Customize <Family Caption> Datasheet** window to show the label again.

4. On the **Customize <Family Caption> Datasheet** window, click the 💾 button.

   Your changes are saved.

Showing Hidden Item Labels

If you have hidden an item's label on a custom-layout datasheet, you can use the following instructions to show the label again.

To show a hidden label on a custom-layout datasheet:

1. In the Meridium APM Framework, open or create a record that uses a custom-layout datasheet containing a hidden label that you want to show.

2. Access the **Customize <Family Caption> Datasheet** window.

3. In the datasheet, right-click the field or group whose hidden label you want to show, and click **Show Text**.

   The label is added to the datasheet.

   > ⓘ **Note:** If you show the label by mistake, you can click the button on the **Customize <Family Caption> Datasheet** window to hide the label again.

4. On the **Customize <Family Caption> Datasheet** window, click the 🖫 button.

   Your changes are saved.

Specifying Where an Item's Label Will Appear

To specify where an item's label will appear on a custom-layout datasheet:

1. In the Meridium APM Framework, open or create a record that uses a custom-layout datasheet containing a label that you want to move.

2. Access the **Customize <Family Caption> Datasheet** window.

3. In the datasheet, right-click the item whose label you want to move.

   A shortcut menu appears.

4. On the shortcut menu, point to **Text Position**, and then click one of the following options:

   - **Top:** Moves the label above the item.

- **Bottom:** Moves the label below the item.

- **Left:** Moves the label to the left of the item.

- **Right:** Moves the label to the right of the item.

   The label moves to the selected location.

5. On the **Customize <Family Caption> Datasheet** window, click the 🖫 button.

   Your changes are saved.

Creating a Group of Fields

> ⓘ **Hint:** You can group only fields that appear in consecutive order on the datasheet. If you want to group a set of fields that are not positioned next to each other, move the fields appropriately before attempting to group them.

To create a group of fields on a custom-layout datasheet:

1. In the Meridium APM Framework, open or create a record that uses a custom-layout datasheet containing fields that you want to group.

2. Access the **Customize <Family Caption> Datasheet** window.

3. In the datasheet, using the Ctrl or Shift key, select the fields that you want to group.

4. Right-click any of the selected fields, and click **Group**.

   The fields are grouped. The group label is set by default to **item<n>**, where <n> is a number representing the number of groups that exist in the datasheet. Note that if this is the first group that exists, the label will be item0.

5. Rename the group as appropriate.

6. On the **Customize <Family Caption> Datasheet** window, click the 💾 button.

   Your changes are saved.

Adding a Field to an Existing Group

To add a field to an existing group on a custom-layout datasheet:

1. In the Meridium APM Framework, open or create a record that uses a custom-layout datasheet containing a field that you want to add to an existing group.

2. Access the **Customize <Family Caption> Datasheet** window.

3. In the datasheet, select the field that you want to add to an existing group. Only active fields can be added to custom-layout datasheets.

4. Drag the selected field to the desired group.

   The field is added to the group.

5. On the **Customize <Family Caption> Datasheet** window, click the 💾 button.

   Your changes are saved.

Removing a Field from a Group

To remove a field from a group on a custom-layout datasheet:

1. In the Meridium APM Framework, open or create a record that uses a custom-layout datasheet containing a field you want to remove from a group.

2. Access the **Customize <Family Caption> Datasheet** window.

3. In the datasheet, select the field that you want to remove from a group.

4. Drag the selected field to the desired location outside of the group.

   The field is removed from the group.

5. On the **Customize <Family Caption> Datasheet** window, click the 💾 button.

   Your changes are saved.
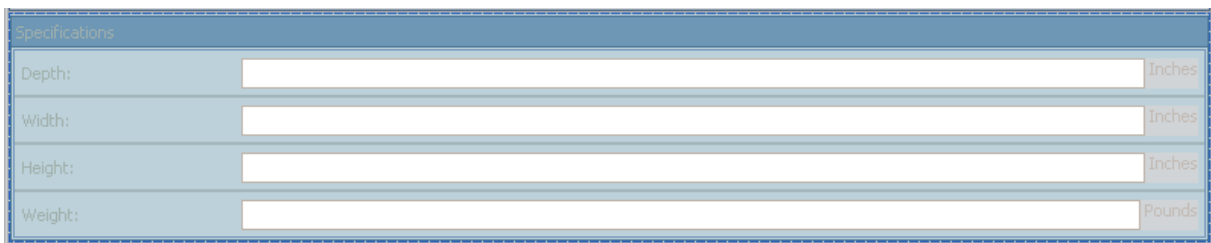
Creating a Tab from an Existing Group of Fields

> ⓘ **Note:** Before you can create a tab on a custom-layout datasheet, you must create a group of fields.

To create a tab from an existing group of fields on a custom-layout datasheet:

1. In the Meridium APM Framework, open or create a record that uses a custom-lay-out datasheet to which you want to add a tab.

2. Access the **Customize <Family Caption> Datasheet** window.

3. In the datasheet, select the group that you want to use for creating a tab.
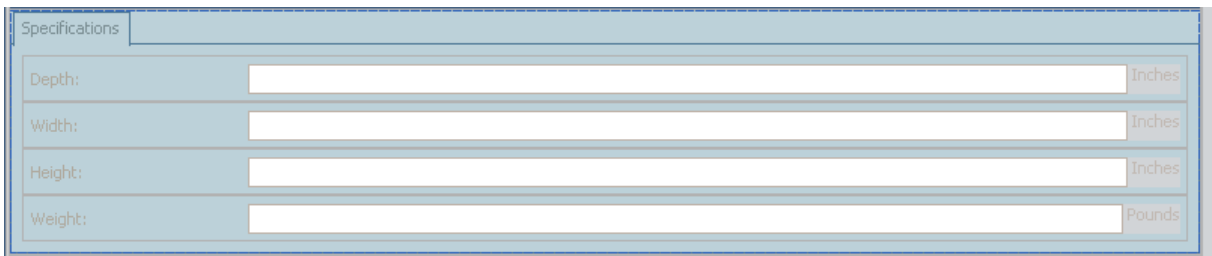
   In the following image, the **Specifications** group is selected.

4. Right-click the selected group, and click **Create TabbedGroup**.

   The group becomes a tab.

5. On the **Customize <Family Caption> Datasheet** window, click the 💾 button.

   Your changes are saved.

Creating a Tab from Scratch

> ⓘ **Note:** To create a tab from scratch on a custom-layout datasheet, at least one tab must already exist on the datasheet. This means that before you can create a tab from scratch, you must create at least one tab from a group of fields.

To create a tab from scratch on a custom-layout datasheet:

1. In the Meridium APM Framework, open or create a record that uses a custom-layout datasheet to which you want to add a tab.

2. Access the **Customize <Family Caption> Datasheet** window.

3. In the datasheet, locate the existing tab next to which you want to add a new tab.

4. Right-click the area to the right of the selected tab, and click **Add Tab**.

   A new tab appears to the right of the existing tab.
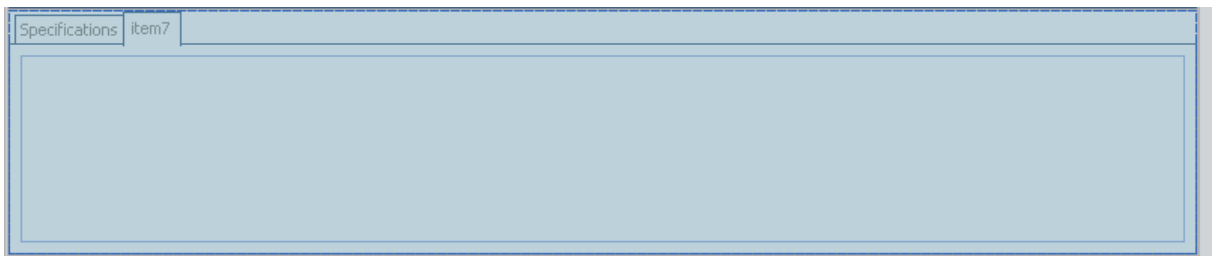
5. Rename the tab as desired.

6. Move the tab as desired.

7. On the **Customize <Family Caption> Datasheet** window, click the 💾 button.

   Your changes are saved.

Undoing an Action

If you make a change to a datasheet configuration, you can undo it if needed.

To undo a change to a custom-layout datasheet configuration:

1. In the Meridium APM Framework, after you make a change to a custom-layout datasheet that you want to undo, on the **Customize <Family Caption> Datasheet** window, click the button.

   The last change that you made is undone.

2. Repeat step 1 as many times as needed to undo additional changes that you made.

Redoing an Action

If you undo an action when configuring a custom-layout datasheet, you can redo the action if needed. For example, if you modify a value and undo the modification, you can re-add the value if desired.

To redo an action when configuring a custom-layout datasheet:

1. In the Meridium APM Framework, after you undo a change to a custom-layout datasheet that you want to redo, on the **Customize <Family Caption> Datasheet** window, click the button.

   The last change that you made is redone.

2. Repeat step 1 as many times as needed to redo additional changes that have been undone.

Exporting a Datasheet Configuration

After you have configured a custom-layout datasheet, you can export the configuration so that it can be imported into another database that contains the same family.

> ⓘ **Hint:** You can also export a datasheet configuration via the Configuration Manager. If you do so, the user who imports the datasheet configuration will not need to save it after the import procedure is completed. If you export the datasheet configuration using the following instructions, however, the user who imports it*will* need to save the configuration as part of the import procedure.

To export a custom-layout datasheet configuration:

1. In the Meridium APM Framework, open or create a record that uses a custom-layout datasheet configuration that you want to export.

2. Access the **Customize <Family Caption> Datasheet** window.

3. On the **Customize <Family Caption> Datasheet** window, click the 💾 button.

   The **Save As** dialog box appears.

4. Navigate to the location in which you want to save the export file, type a name for the file, and click the **Save** button.

   The datasheet configuration is saved to the selected location as an XML file. You can now use this file to import the datasheet configuration into another database that contains the same family.

Importing a Datasheet Configuration

After a datasheet configuration has been exported, you can import the configuration into another database, assuming that the following conditions have been met:

- The target database contains the same family for which the datasheet was configured in the source database.
- A custom-layout datasheet has been defined for this family in the target database.
- The Security User performing the import is a member of the Security Group that is associated with the custom-layout datasheet in the target database.

To import a custom-layout datasheet configuration:

1. In the Meridium APM Framework, open or create a record that uses a custom-layout datasheet whose configuration you want to import.

2. Access the **Customize <Family Caption> Datasheet** window.

3. On the **Customize <Family Caption> Datasheet** window, click the 📂 button.

   The **Open** dialog box appears.

4. Navigate to the location in which the desired datasheet configuration file exists, select the file, and click the **Open** button.

   The datasheet is updated to reflect the imported configuration.

5. On the **Customize <Family Caption> Datasheet** window, click the 💾 button.

   The datasheet configuration is saved.

Saving a Datasheet Configuration

After you make changes to a custom-layout datasheet configuration, you will need to save the configuration so that other users that use the same database can see the updated configuration.

Note that if you save a datasheet configuration as a Security User with a Culture setting *other than* the default setting, English, only Security Users with that Culture setting will be able to see that configuration. For example, if you log in to the Meridium APM Framework application as a Security User with a Spanish Culture setting, uses without a Spanish Culture setting will not see the datasheet configuration that you have created.

In other words, that Culture setting that is used to save a datasheet configuration is required by any Security Users who wants to see that configuration.

To save a custom-layout datasheet configuration:

- In the Meridium APM Framework, after you configure a custom-layout datasheet, on the **Customize <Family Caption> Datasheet** window, click the **Save** button.

  Your changes are saved.

Deleting Datasheets

To delete a datasheet:

1. In the Configuration Manager, access the **Datasheet Builder** for the family to which the datasheet belongs.

2. In the **Manage Datasheets for <Family ID>** section, right-click the row containing the datasheet that you want to delete, and choose **Delete Datasheet** on the short-cut menu.

   A confirmation message appears, asking if you really want to delete the datasheet.

3. Click the **Yes** button to confirm the deletion.

What Are Rules?

In Meridium APM, *rules* consist of code that determines how records in the Meridium APM database will behave under specific conditions. Rule code is written in *Visual Basic.Net (VB.Net)*, a programming language that is compatible with the language in which the Meridium APM applications are written, allowing the Meridium APM rules to plug in to the Meridium APM software. In this way, you can specify that when certain actions are taken in Meridium APM, certain rules should be executed.

The purpose of writing business rules for a family is to control how records in that family will behave when a user works with those records in the Meridium APM Framework application. The rule code itself can be stored in two locations:

- Within the family rule project itself.
- Within the Rules Library.

Rules can range from very simple to highly complex. For very simple field-level rules, the Configuration Manager application provides a builder for generating rule code based upon inputs that you provide through the interface. This tool helps simplify the task of writing *field-level rules* within family rule projects.

For more complex rules, the Configuration Manager application provides access to Microsoft Visual Studio for Applications (VSTA), which gives you direct access to the VB.Net code. If you have sufficient knowledge of writing VB.Net code, VSTA is a powerful tool that lets you customize your system to suit the specific needs of your organization.

Writing complex rules requires knowledge of VB.Net that exceeds the scope of this documentation. The purpose of this documentation is to explain the basic structure of family rule projects and to describe the tools that are available in the Configuration Manager to help you write rules. In addition, we provide some basic information about rule code itself to help you understand and navigate your existing rules and some basic examples of custom rule code.

> ⚠ IMPORTANT: Modifying rules without the proper knowledge and expertise could cause your system to work improperly. Before you begin modifying business rules, you may want to export the existing rules so that you can revert to them if needed.

•

Using the Meridium Rules Editor (VSTA)

Microsoft Visual Studio Tools for Applications (VSTA) is the tool that Meridium APM uses to display the **Meridium APM Rules Editor**, which serves as the interface where you can view and modify all the VB.Net code within a given rule project. This tool is accessible via the Configuration Manager and serves as the interface for viewing and modifying both [family rule projects](#) and [Rules Library Projects](#). This means that you can access VSTA in several ways, depending upon which rule code you want to edit.

In other words, you will use VSTA when you choose to:

- [Open a family rule project](#).
- [Create a new rule project](#) or [modify an existing rule project](#) using the Rules Library.

This documentation does not provide a great amount of detail on using VSTA itself. A Help system is available in VSTA and can provide you with more information. One important feature to note is that you can compile rule projects by clicking **Build [Project Name]** on the **Build** menu. When you compile the source code, it will be saved automatically. You will need to compile the rule code any time you make a change to it in order for your change to take effect. You can wait and compile the entire rule project after you have made all the desired changes to the project.

Overview of Rule Code Storage Options

Custom family- and field-level rules can be stored in two places:

- Within the [family rule project](#) itself.
- Within the [Rules Library](#).

Storing family rules within the family rule project itself means that the actual rule code is stored in the family and field classes within a family rule project. This form of rule code storage is acceptable but can be cumbersome as it can result in a large amount of rule code that must be maintained on a family-by-family basis. In addition, rule code that is stored within family rule projects applies *only* to one family. To apply the same rule code to another family, you would have to copy the rule code and paste it into another family rule project.

The Rules Library, on the other hand, stores rule code in projects that can be *referenced* from family rule projects. In this way, you can store the actual VB.Net code in one central location and then apply it to multiple families. This method of rule code storage offers many advantages, including limiting the amount of code that must be maintained and increasing the ease and efficiency with which that code can be applied to other families.

The Rules Library, however, also imposes some limitations. To take advantage of the *exactsame* rule code across multiple families, you would need to have multiple families that should behave *exactly* the same way. It is more likely, however, that you will have multiple families that should behave in *similar* ways.

For this reason, you will probably want to use a combination of the two rule code storage methods. In the Rules Library, you can store code that will serve as the foundation upon which family- and field-level rules are built. After referencing that code from a family rule project, you can extend the rule through family-level and field-level customization.

Rule Projects and Their Structure

A *rule project* is a container for rule code. Meridium APM uses two types of rule projects:

- **Family Rule Project**: Stores all the rules for a given family.
- **Rules Library Project**: Stores all the rules that exist for that project.

Rule projects contain *references* and *files*, also called code items. The files contain rules that are defined for that project. The file structure for family projects is determined by the content of the family. The file structure of a Rules Library project is determined by the project owner and can be customized as necessary to meet the requirements of the project.

Classes

A *class* is a VB.NET element that serves as a container for storing objects. Classes are defined within the files that exist in a [rule project](#).

- In [family rule projects](#), Meridium APM automatically defines classes within the files that exist for the family itself and the fields within that family.
- In [baseline Rules Library projects](#), classes serve as containers for the code within those projects.
- In custom Rules Library projects, you can define your own classes as needed. When you create a new project, one file will be created within that project, and within that file, a default class will be defined.

Any VB.NET class can be [inherited](#) from another class so that the rules defined in one class can be reused as often as needed.

The following code excerpt shows an example of the default class *Class1* that is created when you create a new Rules Library project:

```
------------------------------------------------------------------

Option Strict On
Option Explicit On

Imports Meridium.Core.DataManager
Imports Meridium.Core.DataManager.Customization
Imports Meridium.Core.Internals
Imports Meridium.Core.Metadata
Imports Meridium.Core.Security.ApplicationUser
Imports Meridium.Core.Uom
Imports System
Imports System.Xml

Public Class Class1

    Private Sub New()
       MyBase.New
       '
       ' TODO: Add constructor logic here
       '
    End Sub
End Class
```

In this example, the text in red identifies the code that defines the class; the code in between these two lines of text represents all the objects that belong to the class.

Functions

A *function* is a block of rule code that defines a Function procedure. Functions can be defined to invoke specific behaviors. [The standard field-level rules that are available in Meridium APM are defined through functions](). In the following example, the *IsRequired* function is shown in red.

```
----------------------------------------------------------------

Option Strict On
Option Explicit On

Imports Meridium.Core.DataManager
Imports Meridium.Core.DataManager.Customization
Imports Meridium.Core.Internals
Imports Meridium.Core.Metadata
Imports Meridium.Core.Security.ApplicationUser
Imports Meridium.Core.Uom
Imports System
Imports System.Xml

Public Class Class1

        Public Overrides Function IsRequired() As Boolean
                Return True
        End Function

End Class
```

Inheritance

Inheritances is a feature of VB.Net classes that allows one class to use the behaviors defined in another class.

- The class that is inherited is considered the *base class* and serves as foundation for the functionality of the class that inherits it.
- Any class that inherits *from* another class is considered a derived class.

All the functions and behaviors defined in the base class are automatically applied to derived class. Within the derived class, code can be written to extend or override specific functions defined in the base class. Inheritance allows you to create new classes based on existing classes and is an important component of the Rules Library and baseline business rule storage.

Inheritance is achieved through an *Inherits statement* in the derived class. For instance in the following example, the class **MI_RCA_ANALY_COST_NBR** (the derived class) inherits the class **EntityFieldCustomization** (the base class).

Public Class MI_RCA_ANALY_COST_NBR

    Inherits EntityFieldCustomization

About Family Rule Projects

A *family rule project* is an organizational unit that provides the coding infrastructure within which the rules for that family will be written and stored.

- For baseline families, a family rule project exists for any family for which family or field-level rules have been defined *within that specific family*. For baseline sub-families that inherit all their behaviors from higher-level families, a family rule project will not exist.

- For custom families, the family rule project is created the first time you access the family-level or field-level rules for the family.

Family rules can be viewed and managed in the **Meridium APM Rules Editor**. When you open the rule project for a family using the **Manage Family Business Rules** link, the family rule project will be selected by default in the **Project Explorer** pane. In the following image, the rule project for the *Motor* family is selected. Below the project name is a list of all the items that are currently included in the project.
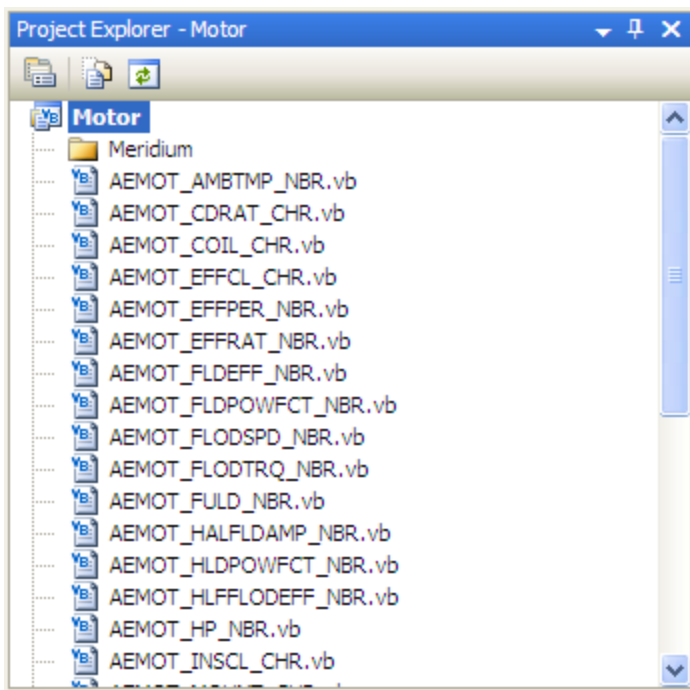
By default, the content of the project will consist of references and files, identified by the [icon] icon. Each family rule project contains a file for the family itself and a file for each field within the family. A file will be created for each field that is defined directly within that family *and* each field that is spread down from a family where the field is allowed to be customized at the sublevel. As you add fields to a family, new files will be added to the family rule project. You can double-click any file in the **Project Explorer** pane to view and edit the code that is stored within it.

Within each file, a *class* is defined for the corresponding family or field. Each class serves as the organizational unit within which actual family-level and field-level

customization code exists. The name of the file matches the name of the class defined within it, which in turn corresponds to the field ID or family ID (i.e., not the family or field caption). For example, in the Motor project shown above, the class AEMOT_ AMBTMP_NBR is defined within the file AEMOT_AMBTMP_NBR and stores the rule code for the field AEMOT_AMBTMP_NBR.

```
<MetadataField("AEMOT_AMBTMP_NBR")> _
Public Class AEMOT_AMBTMP_NBR
    Inherits Meridium.Core.DataManager.Customization.EntityFieldCustomization

    Public Sub New(ByVal record As Meridium.Core.DataManager.DataRecord, ByVal field
As Meridium.Core.DataManager.DataField)
        MyBase.New(record, field)
    End Sub
End Class
```

> **Note:** Changing a field's ID when the field already has a VB.Net file will cause the Meridium APM system to create a new file based upon the new ID. The old file will not be deleted but will be disconnected from the field.

If needed, you can open multiple rule projects at a time in VSTA. When you do so, each rule project will appear as a root-level entry in the **Project Explorer** pane. To open multiple projects, leave VSTA open, return to the Configuration Manager main window, and open the rule project for another family.

Note that when you access the family rule project for a *baseline* Meridium APM family, the corresponding baseline Rules Library project will also be displayed in the **Project Explorer** pane. The baseline Rules Library project has the same name as the family rule project with **_Base** appended to it. You cannot modify the baseline Rules Library project, but it is displayed so that you can easily view and debug the baseline rule code to understand the baseline family and field rules.

Accessing a Family Rule Project

You can access family rule projects at the *family* level or the *field* level. Keep in mind that in either case you are opening the *same* project. The difference is that depending upon your point of access, the system will highlight a different node in the **Project Explorer** pane. Once the project is open, however, you can navigate anywhere within the entire project.

The following instructions provide details on accessing the family rule project at the *family* level. You can access the family rule project at the field level via the <u>View/Edit Field Properties</u> window.

To access a family rule project:

- In the Configuration Manager, in the **Entity Family/Relationship Family** pane (on either the **Entity Family** or **Relationship Family** tab), select the desired family, and then in the **Tasks** area of the Configuration Manager main window, click the **Manage Family Business Rules** link.

  The **Meridium APM Rules Editor** (VSTA) appears, displaying the family rule project, and the root level of the project is selected.
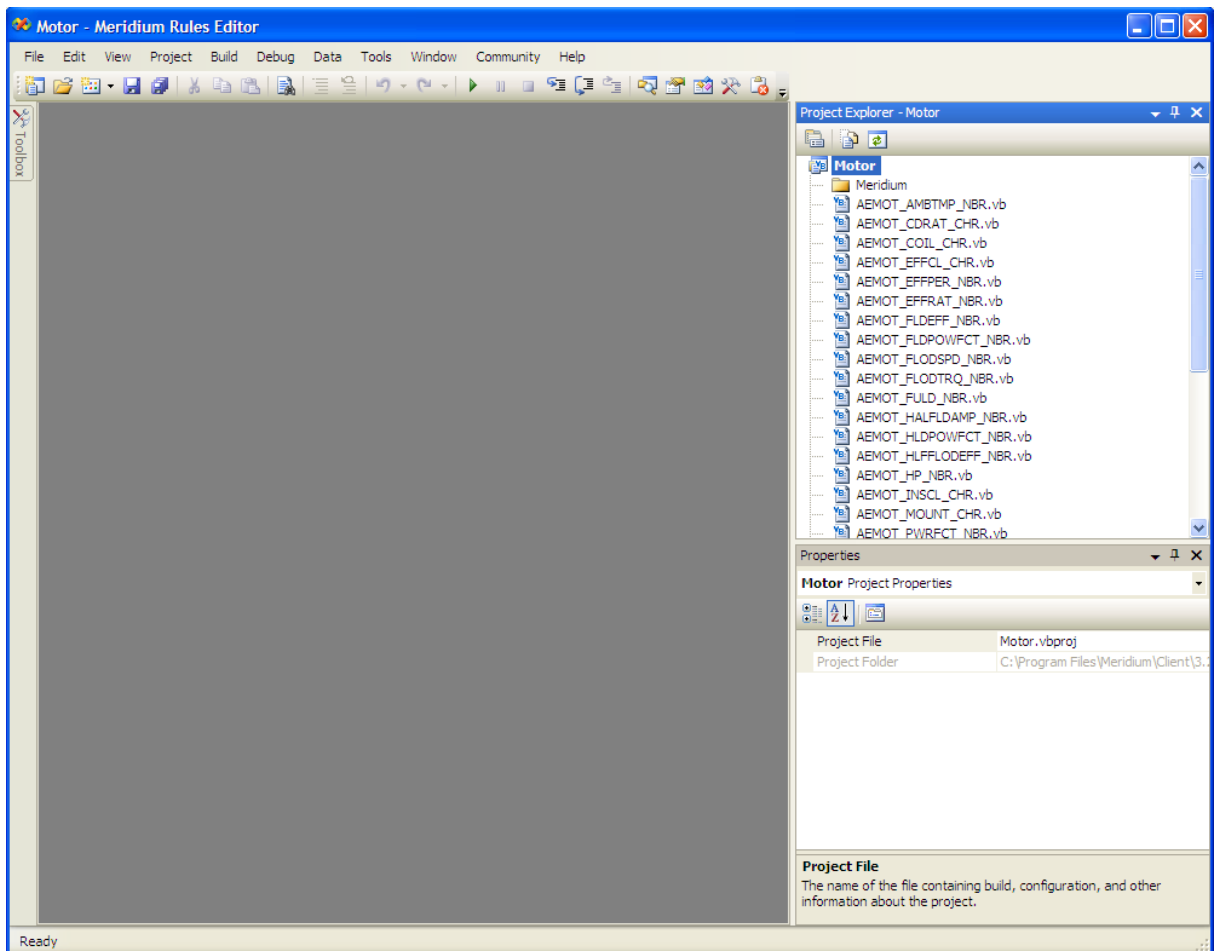
What Are Family-Level Rules?

Whereas [field-level rules](#) control the behavior of specific *fields* within a record, family-level rules reside within the [code item that represents an entity family](#) and control actions performed against the entire record. Family-level rules provide you with a great deal of flexibility in determining how records in your database will behave. Some of the most common uses of family-level rules include:

- Managing record links.
- Calculating values.
- Sending email notifications.

Meridium APM supports the family-level rule types described in the following table.

Rule Type	Function	Stores logic that is executed...
Before Insert	Public Overrides Sub BeforeInsert() MyBase.BeforeInsert() End Sub	Before a record is created.
After Insert	Public Overrides Sub AfterInsert() MyBase.AfterInsert() End Sub	After a record is created.
Before Update	Public Overrides Sub BeforeUpdate() MyBase.BeforeUpdate() End Sub	After changes have been made to a record, before those changes are saved to the database.
After Update	Public Overrides Sub AfterUpdate() MyBase.AfterUpdate() End Sub	After changes to a record have been saved.

•

Rule Type	Function	Stores logic that is executed...
Before Delete	Public Overrides Sub BeforeDelete() MyBase.BeforeDelete() End Sub	Before a record is deleted.
After Delete	Public Overrides Sub After-Delete() MyBase.AfterDelete() End Sub	After a record has been deleted.

Unlike field-level rules, which can be generated using the Rules Wizard, family-level rules must be created by developing custom code. You can use function templates provided in this table and insert them into the family-level code items to create the framework for developing each type of rule.

•

Example: Linking Two Records Using an AfterInsert Rule

One of the most common uses of an AfterInsert rule is to create a link between two records after a new record is created. Consider an example where you have a Work Order family that is related to the Equipment family through the Has Maintenance relationship. When you create a Work Order record to represent work performed on a piece of equipment, you want to link it to the Equipment record that represents the piece of equipment on which the work will be performed. You can accomplish this by creating a family-level rule for the Work Order family.

The following code shows an example of an AfterInsert rule that you could write to create a link between the Equipment record and the Work Order record when the Work Order record is created. You would insert this rule into the family-level code item for the Work Order family. This example assumes that:

- The Work Order family contains the Equipment ID (EWORK_EQUIP_ID_CHR) field, which is populated with the Equipment ID of the Equipment record for which the Work Order record is created.

- The Equipment family contains the Equipment ID (MI_EQUIP000_EQUIP_ID_C) field, which stores a unique value that identifies each Equipment record.

- The Equipment field in the Work Order family has a Valid Values rule defined for it that populates it with a list of Equipment IDs of the Equipment records that exist in the database. Note that this condition is not required and is not represented by the following rule code. Defining this rule, however, will help ensure that the Equipment ID field contains a valid Equipment ID, which is necessary for this rule to work properly.

```
-------------------------------------------------------------------------

Dim ePred As Entity
Dim eSucc As Entity
Dim oRs As Rowset
Dim Reader As Meridium.Core.Metadata.MetadataReader
Dim FamilyObject As Meridium.Core.Metadata.Family
Dim Rel As Relationship
Dim Cmd As Command

Dim lRelationshipKey As Long
Dim lPredecessorKey As Long
Dim lPredFAM_Key As Long
Dim sSQL As String

Try

        If Not Convert.IsDBNull(CurrentEntity.Fields("EWORK_ASSET_ID_CHR").Value) Then

                Reader = Meridium.Core.Metadata.MetadataReader.CreateInstance(CurrentEntity.ApplicationUser)
                FamilyObject = Reader.RelationshipFamilies.FindItem("Has Maintenance")
```

```
        lRelationshipKey = FamilyObject.Key

        Reader = Meridium.Core.Metadata.MetadataReader.CreateInstance(Cur-
        rentEntity.ApplicationUser)
        FamilyObject = Reader.EntityFamilies.FindItem("Equipment")
        lPredFAM_Key = FamilyObject.Key

        sSQL = "SELECT ENTY_KEY FROM [Equipment] WHERE [Equipment].[MI_
        EQUIP000_EQUIP_ID_C] = '" & CStr(CurrentEntity.Fields("EWORK_EQUIP_ID_
        CHR").Value) & "'"
        Cmd = New Command(CurrentEntity.ApplicationUser)
        Cmd.CommandText = sSQL
        oRs = Cmd.Execute(RowsetMode.RawData)

        If oRs.Rows.Count > 0 Then
              lPredecessorKey = CLng(oRs.Rows(0)(oRs.Columns(0)))

              ePred = Entity.ExistingEntity(CurrentEntity.ApplicationUser, lPre-
              decessorKey, lPredFAM_Key)
              eSucc = Entity.ExistingEntity(CurrentEntity.ApplicationUser, Cur-
              rentEntity.Key,CurrentEntity.Family.Key)

              Rel = Meridium.Core.DataManager.Relationship.NewRelationship(Cur-
              rentEntity.ApplicationUser, lRelationshipKey, ePred, eSucc)
              Rel.Update(UpdateMode.Manual)

        Else
              Throw New Meridium APMException("Could not find Equipment
              record with this ID: " & CStr(CurrentEntity.Fields("EREPAI_ASSET_ID_
              CHR").Value))
        End If

    End If

Catch e As Exception

    Throw New Meridium APMException("Error Occurred in Work Order AfterInsert
    rule" & e.Message)

End Try
```

About Field-Level Rules

Field-level rules define how a field will behave under certain circumstances. The field-level rules for a given field are stored within the family rule project for the family to which the field belongs. Each family rule project contains a code item for each field that exists within the family. The rules for a given field are stored in the file that corresponds to that field.

> 🛈 **Note:** Code items will not exist for fields that have been spread down from a higher-level family and are configured at the subfamily level to inherit rules from the source family. In this case, the code item that exists in the family rule project of the *source* family will be used for defining and executing rules at the subfamily level. If the subfamily is configured not to inherit rules from the source family, a code item will exist within the family rule project of the subfamily and will be used for defining and executing rules at the subfamily level.

The following types of rules can be defined for each family field:

- **Required:** Determines whether or not a value must be entered into a field before a record can be saved in the family.

- **Validation:** Lets you define criteria that will be used to validate values that are entered into a field. By creating a Validation rule, you can force the values in a field to conform to specified limits or criteria that are considered valid.

- **Valid Values:** Lets you define a list of values that will be available for selection in the field. Users will be able to select any value from those defined in the list of Valid Values.

- **Default Value:** Defines the default value that will be provided for a field. When a user creates a new record in the Record Manager, the default value will be provided automatically. Users can accept the default value or specify a different value.

- **Disabled:** Determines when, if ever, the field will be disabled, or locked from editing. Disabled rules determine whether or not users can modify the value in the field.

- **Format:** Determines the formatting that will be applied to values entered into a field.

- **Formula:** Calculates the value in the field using a formula that has been specified through rules.

  > 🛈 **Note:** Formula rules can be defined *only* for Formula fields, which have properties that are different from those of non-formula fields.

Field-level rules can be defined in two ways:

- You can use the Rules Wizard to choose from a set of standard, pre-defined options, and the Meridium APM system will generate the necessary rule code for you. This method makes defining field-level rules easy, but it imposes some

limitations. For example, you are limited to the options that are available for selection through the interface, and there is no facility for specifying how fields should interact with other fields and families.

- [You can write rule code yourself using custom code.](#) This method allows more flexibility in the type of rules you can define but requires more expertise. You can define field-level rules in this way only if you know how to write rule code.

> ℹ **Note:** Formula rules cannot be defined through the Rules Wizard. They can be defined only through custom code.

Regardless of how field-level rules are created, they are stored in the same place: in the field-specific classes that exist in family rule projects.

Accessing Field-Level Rules

Using the Configuration Manager, you can access the field-level rules that exist for a family in two ways:

1. By opening the family rule project and then opening the code item associated with the field whose rules you want to view.

2. By clicking the **Edit Rule** link in the **Field Rules** area of the **View/Edit Field Properties window**. Note that this link will be enabled *only* if custom rules exist for a given field.

Meridium APM Default vs. Custom Rules

Field-level rules are stored in field-specific code items within family rule projects. Depending upon the content of the field-specific code item, each rule type (Required, Validation, Valid Values, Default Value, Disabled, and Format) is designated as either:

- **Meridium APM Default:** The default Meridium APM behavior is in place. In other words, the function for that rule type is not defined within the field-specific code item.

  - For the fields in *baseline* Meridium APM families, rules are defined in the Rules Library and be inherited at the field level. For baseline families, *Meridium APM Default* always indicates that the baseline *inheritance* is in place. Depending upon the content of the Rules Library project that is inherited, the field may or may not have any actual behavior.

  - For fields in *custom* families (i.e., families that are added to the Meridium APM baseline database), rules will not exist by default. So, in the case of custom families and fields, *Meridium APM Default* always means that *no rules* exist and none will be executed.

- **Custom:** Custom rule code exists. In other words, the function for that behavior type IS defined within the field-specific code item. Note that Meridium APM makes no distinction between rule code that was generated by the Rules Wizard or developed manually by editing the rule project directly. If the function exists within the project, the behavior is considered to be *Custom*.

Field-level rules are identified as either Meridium APM Default or Custom in two places in the Configuration Manager application:

- In the **Field Rules** area on the **View/Edit Field Properties** window. In the following image, notice that the Required rule type is designated as *Custom* and all other rule types are designated as *Meridium APM Default*.
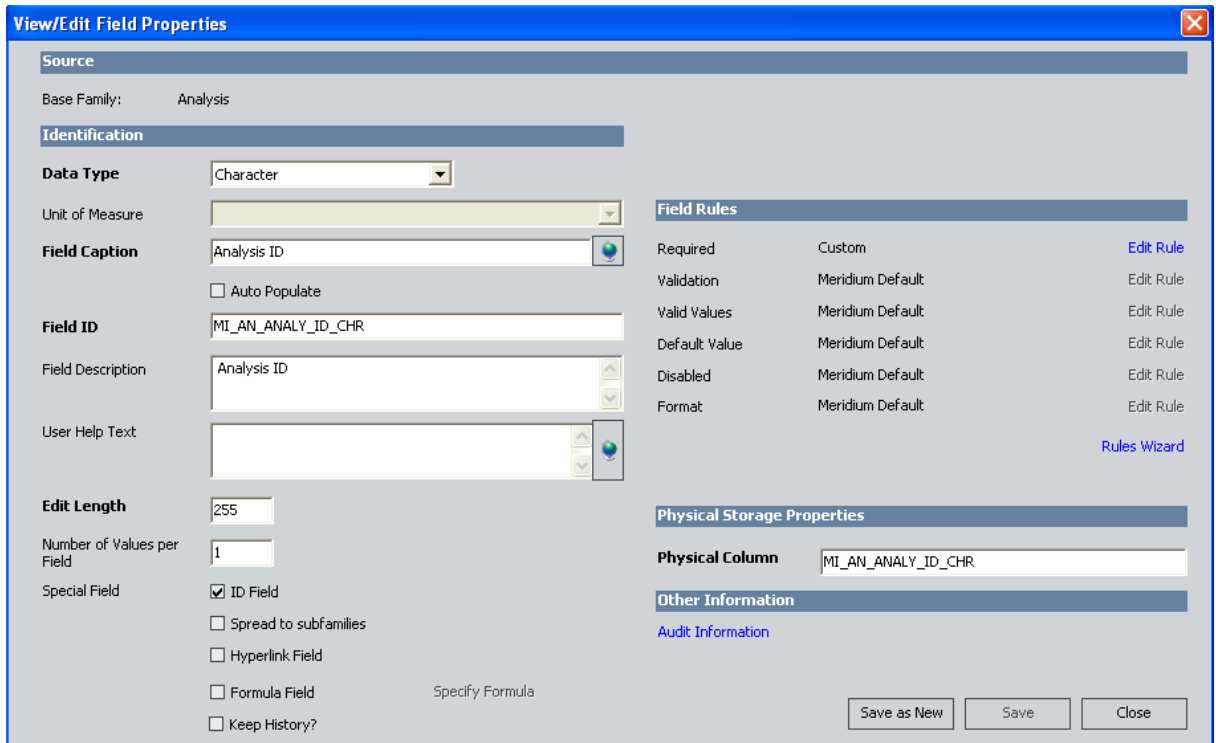
- In the lists on the **Rule Details** screen in the Rules Wizard. In the following image, notice that *Custom* and *Meridium APM Default* appear in the list for the Required rule. *Custom* will appear in the list only if custom code exists for that rule type. *Meridium APM Default* is always included in the list can be used to maintain the baseline functionality or to revert to the default behavior if custom code already exists.
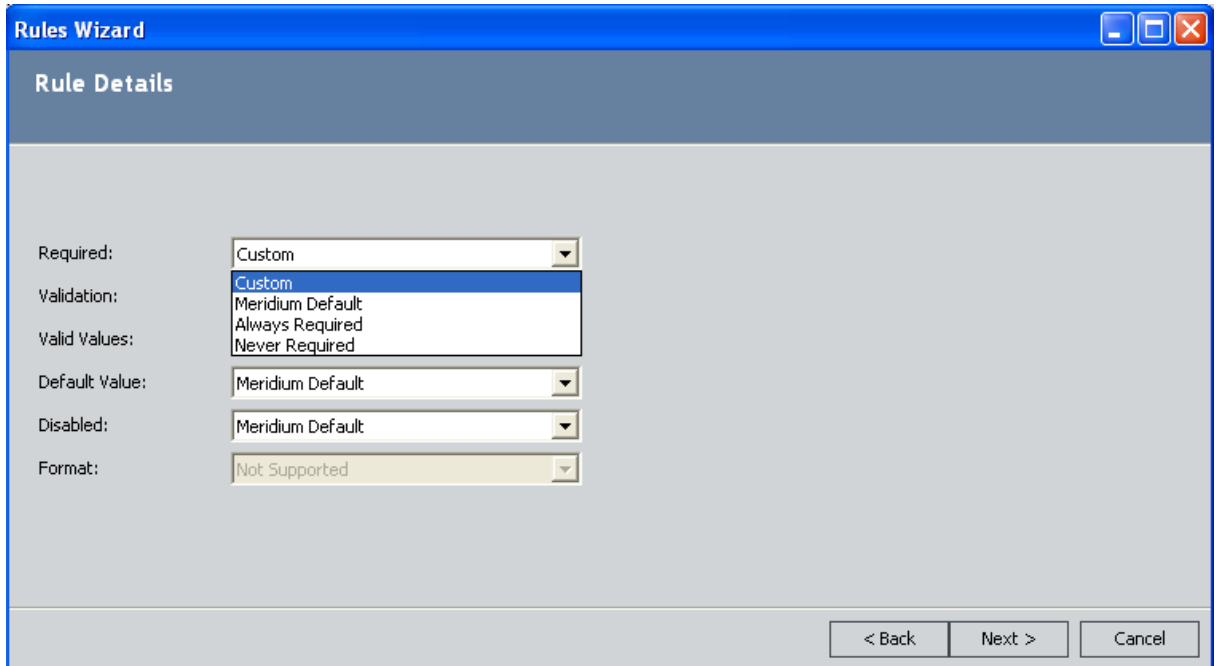
About the Rules Wizard

The Rules Wizard provides you with a tool for generating simple, standard field-level rules. Using the Rules Wizard, you can generate the following types of rules:

- Required
- Validation
- Valid Values
- Default Value
- Disabled
- Format

> 🗋 **Note:** To develop rules that are more complex than those supported by the Rules Wizard, you will need to modify the field-level rules directly.

Using the Configuration Manager, you can access the Rules Wizard:

- From the **View/Edit Field Properties** window to generate rules for a single field.
- From the **Manage Family Fields** window to generate rules for multiple fields.

> ⚠ **IMPORTANT:** When you define field-level rules through the Rules Wizard, the rules will be compiled automatically. This automatic compilation will compile the *entire* family rule project, including any uncompiled custom code that exists.

Generating Required Rules

Required rules allow you to determine whether or not a value is required in a given field. When a field is required, it means that a value must be entered into that field before a record in the family can be saved. When users create a new record in Meridium APM, they must specify a value for each required field. If they do not, an error message will be displayed when they attempt to save the record. You can designate a field as required by defining a Required rule for it.

> ⓘ **Hint:** Via the Configuration Manager, you can choose the background color that will appear on datasheets for required fields. The background color will be applied to all fields where rules exist to make the field required.

To generate a Required rule for a field:

1. In the Configuration Manager, on the **View/Edit Field Properties window**, in the **Field Rules** section, click the **Rules Wizard** link.

   The **Rules Wizard** appears, displaying the **Rule Details** screen.

2. In the **Required** list, select one of the following options:

   - **Meridium APM Default:** The default Meridium APM Required rule for the field.

> 🛈 **Note:** This option will be available only if a custom Required rule already exists. You can use this option to revert back to the Meridium APM Default rule.

- **Always Required:** A value is always required in the field. This means that when a user creates a record for this family, a value will be required in this field.

> ⓘ **Note:** If you set the Required rule for a field to *Always Required*, that field will be spread automatically to all subfamilies of the current family, even if the **Spread to Subfamilies** check box is not selected on the **View/Edit Field Properties** window.

- **Never Required:** A value is never required in the field. By default, fields are not required. If you create a new field, it will not be required unless you set the Required rule to Always Required. The Never Required option is useful for removing the Always Required condition from a field. It is not necessary to set new fields explicitly to Never Required.

> **Note:** The **Custom** option will also appear in the **Required** list if a Required rule already exists for the field, whether that rule was generated via the Rules Wizard or defined manually via the [family rule project](). Choosing the **Custom** option and proceeding through the Rules Wizard will cause the existing field-level rule to remain in tact and not be modified. Selecting any other option, however, will cause the existing field-level rule to be overwritten by the new rule code that is generated as a result of your selection.

3. Click the **Next** button.

   The **Summary** screen appears, displaying a summary of the rule that will be generated for the field.

4. Click the **Next** button.

   The **Generating Rules** screen appears, and the field-level rules are generated. A progress bar appears to indicate the status of the rule-generation process. After the field-level rules have been generated, the family rule project will be compiled automatically. A message above the progress bar on the **Generating Rules** screen will indicate the status of the compilation. After the rules have been generated and compiled, the text **Done** will appear above the progress bar, and the **Finish** button will become enabled.

5. Click the **Finish** button.

   The **Rules Wizard** closes. The rule-generation process is complete.

Generating Validation Rules

Validation rules let you define criteria that will be used to validate values that are entered into a field. By creating a Validation rule, you can force the values in a field to conform to specified limits or criteria that are considered valid. For example, if a Validation rule exists for a field, when a user enters a value into that field, the Meridium APM system will execute the Validation rules to make sure that the value entered is acceptable.

Validation rules can be defined for any field. The standard options described in the following information, however, can be set only for *numeric* and *date* fields. When you access the Rules Wizard for any other field type, the **Validation** list will display the text **Not Supported**. For these field types, Validation rules must be defined via custom rules.

To generate Validation rules for a field:

1. In the Configuration Manager, on the [View/Edit Field Properties window](#), in the **Field Rules** section, click the **Rules Wizard** link.

    The **Rules Wizard** appears, displaying the **Rule Details** screen.

2. In the **Validation** list, select one of the following options:

    - **Meridium APM Default:** The [default Meridium APM Validation rule](#) for the field.

      > ⓘ **Note:** This option will be available only if a custom Validation rule already exists. You can use this option to revert back to the Meridium APM Default rule.

•

- **Greater Than:** The value entered into the field must be greater than the specified value. In the text box that appears to the right of the list, type the value that will be used for performing the validation. If you are defining a Validation rule for a data field, you can select the desired date from the Calendar.

- **Greater Than or Equal To:** The value entered into the field must be greater than or equal to the specified value. In the text box that appears to the right of the list, type the value that will be used for performing the validation. If you are defining a Validation rule for a data field, you can select the desired date from the Calendar.

- **Less Than:** The value entered into the field must be less than the specified value. In the text box that appears to the right of the list, type the value that will be used for performing the validation. If you are defining a Validation rule for a data field, you can select the desired date from the Calendar.

- **Less Than or Equal To:** The value entered into the field must be less than or equal to the specified value. In the text box that appears to the right of the list, type the value that will be used for performing the validation. If you are defining a Validation rule for a data field, you can select the desired date from the Calendar.

- **Between Exclusive:** The value entered into the field must be between the two specified values but *cannot include* the specified values themselves. In the text boxes that appears to the right of the list, type the values that will be used for performing the validation. If you are defining a Validation rule for a data field, you can select the desired date from the Calendar.

- **Between Inclusive:** The value entered into the field must be between the two specified values and *can include* the specified values themselves. In the text boxes that appears to the right of the list, type the values that will be used for performing the validation. If you are defining a Validation rule for a data field, you can select the desired date from the Calendar.

- **Equals:** The value entered into the field must be equal to the specified value. In the text box that appears to the right of the list, type the value that will be used for performing the validation. If you are defining a Validation rule for a data field, you can select the desired date from the Calendar.

> Note: The **Custom** option will also appear in the **Validation** list if a Validation rule already exists for the field, whether that rule was generated via the Rules Wizard or defined manually via the [family rule project](). Choosing the **Custom** option and proceeding through the Rules Wizard will cause the existing field-level rule to remain in tact and not be modified. Selecting any other option, however, will cause the existing field-level rule to be overwritten by the new rule code that is generated as a result of your selection.

3. Click the **Next** button.

   The **Summary** screen appears, displaying a summary of the rule that will be generated for the field.

4. Click the **Next** button.

The **Generating Rules** screen appears, and the field-level rules are generated. A progress bar appears to indicate the status of the rule-generation process. After the field-level rules have been generated, the family rule project will be compiled automatically. A message above the progress bar on the **Generating Rules** screen will indicate the status of the compilation. After the rules have been generated and compiled, the text **Done** will appear above the progress bar, and the **Finish** button will become enabled.

5. Click the **Finish** button.

The **Rules Wizard** closes. The rule-generation process is complete.

About Valid Values Rules

Valid Values rules let you define a list of values that will be available for selection in the field. Users will be able to select any value from those defined in the list of Valid Values. Via the Rules Wizard, you can create three types of valid values lists:

- **Meridium APM Default:** The default Meridium APM Valid Values rule for the field.

  > ⓘ **Note:** This option will be available only if a custom Valid Values rule already exists. You can use this option to revert back to the Meridium APM Default rule.

- **Static List:** A list of static values that you define.

  > ⓘ **Note:** You cannot add null values to a static valid values list using the Rules Wizard. To add null values to a static valid values list, you must modify the family rule project manually.

- **System Code Table:** A list based on a System Code Table, which is a list of values that matches the codes defined in a System Code Table.

> ⓘ **Note:** The **Custom** option will also be available if a Valid Values rule already exists for the field, whether that rule was generated via the Rules Wizard or defined manually via the family rule project. Choosing the **Custom** option and proceeding through the Rules Wizard will cause the existing field-level rule to remain in tact and not be modified. Selecting any other option, however, will cause the existing field-level rule to be overwritten by the new rule code that is generated as a result of your selection.

Note that Valid Values rules should not be configured for text fields. Although the **Valid Values** option is enabled on the **View/Edit Field Properties** window when you are viewing the properties of a text field, if you define Valid Values rule for a text field, the values that you specify do not appear on the datasheet in the Meridium APM Framework application. Instead, users can type any value.
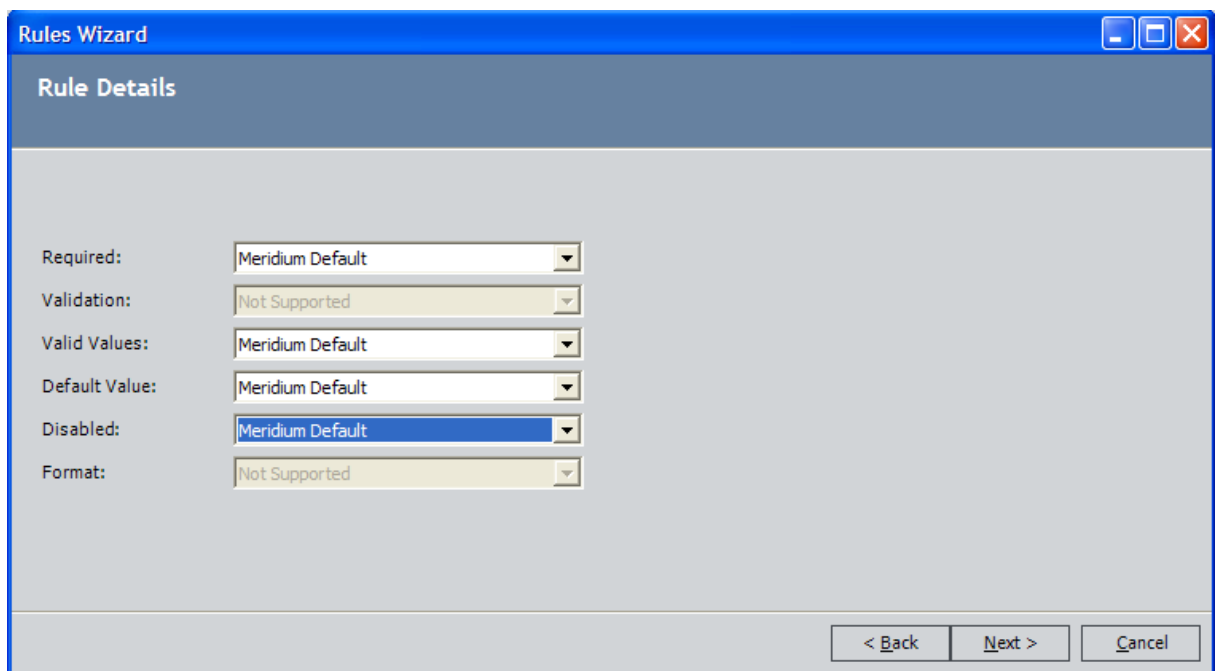
Creating a Static List of Valid Values

The following instructions provide details on configuring a field to contain a static list of values, meaning that the list of values is defined in the rule itself and will never change unless the rule is modified. Note that static Valid Values lists that are created in this way will be *restricted* by default. If desired, you can [modify the rule to make the list unrestricted](#).

To create a static list of valid values:

1. In the Configuration Manager, on the [View/Edit Field Properties window](#), in the **Field Rules** section, click the **Rules Wizard** link.

   The **Rules Wizard** appears, displaying the **Rule Details** screen.

2. In the **Valid Values** list, select **Static List**.

   The **Static List** dialog box appears.

3. In the text box at the top of the dialog box, type the first static value.

   > **Note:** Pipe characters (|) and quotation marks (") should not be used in static Valid Values lists.

4. Click the **Add** button.

   The value is added to the list below the text box.

5. Continue typing values in the text box and clicking the **Add** button until you have added all the desired values. If you need to delete a value, select the value in the list, and click the **Delete** button. Note that:

   - No character-limit validation is performed on the values that you define. When creating a list of static values, be sure that none of the values exceed the maximum character length of the field. Values that exceed the allowed character length will cause the Meridium APM system to generate validation errors.

   - You cannot add a null value to the list via the **Valid Values** window. To add a null value, you must [modify the rule code itself](#).

6. When you have finished adding all the desired values, click the **Next** button.

   The **Summary** screen appears, displaying a summary of the rule that will be generated for the field.

7. Click the **Next** button.

   The **Generating Rules** screen appears, and the field-level rules are generated. A progress bar appears to indicate the status of the rule-generation process. After the field-level rules have been generated, the family rule project will be compiled automatically. A message above the progress bar on the **Generating Rules** screen will indicate the status of the compilation. After the rules have been generated and compiled, the text **Done** will appear above the progress bar, and the **Finish** button will become enabled.

8. Click the **Finish** button.

   The **Rules Wizard** closes. The rule-generation process is complete.

Creating a Valid Values List from a System Code Table
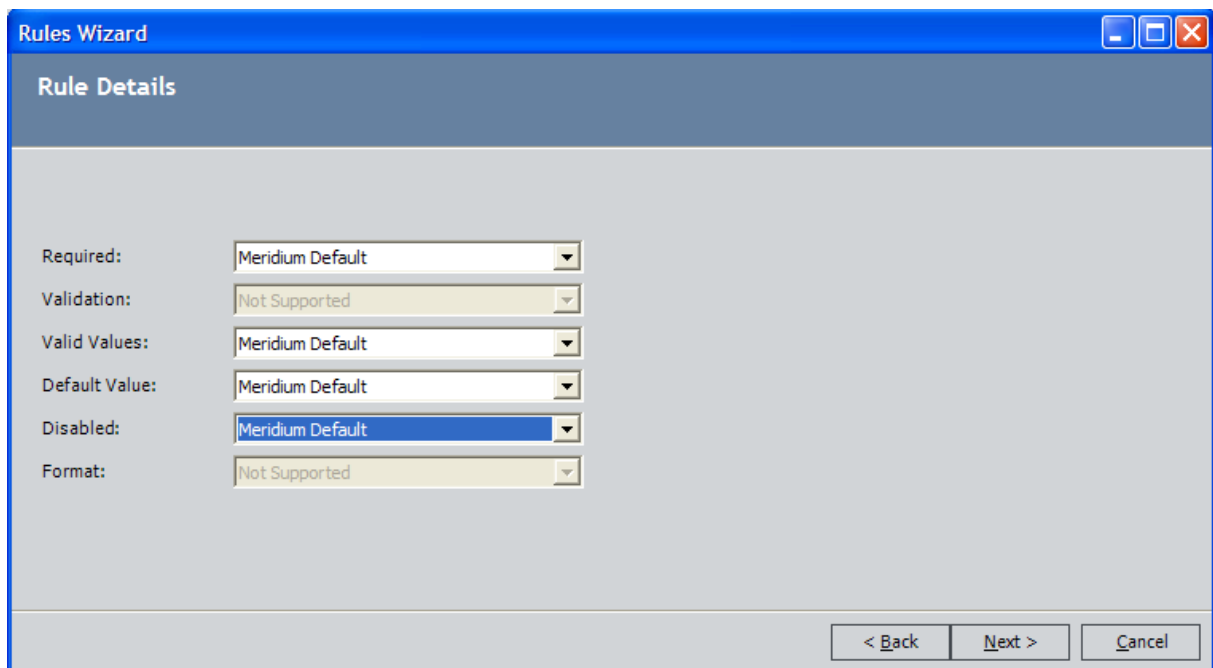
The following instructions provide details on configuring a field to contain a list of System Codes defined in a System Code Table. When you create a Valid Values list in this way, the drop-down list of valid values will display any System Codes that are defined in the specified System Code Table. The list will be updated automatically to reflect changes made to the System Code Table. Note that Valid Values lists created in this way will be *restricted* by default. If desired, you can modify the rule to make the list unrestricted.

> Note: Only *active* System Codes will appear in valid values lists that are constructed from System Code Tables.

To create a valid values list from a System Code Table:

1. In the Configuration Manager, on the **View/Edit Field Properties** window, in the **Field Rules** section, click the **Rules Wizard** link.

   The **Rules Wizard** appears, displaying the **Rule Details** screen.

2. In the **Valid Values** list, select **System Code Table**.

   The **System Code Wizard** appears, displaying the **Select Valid Values System Code Table** screen.
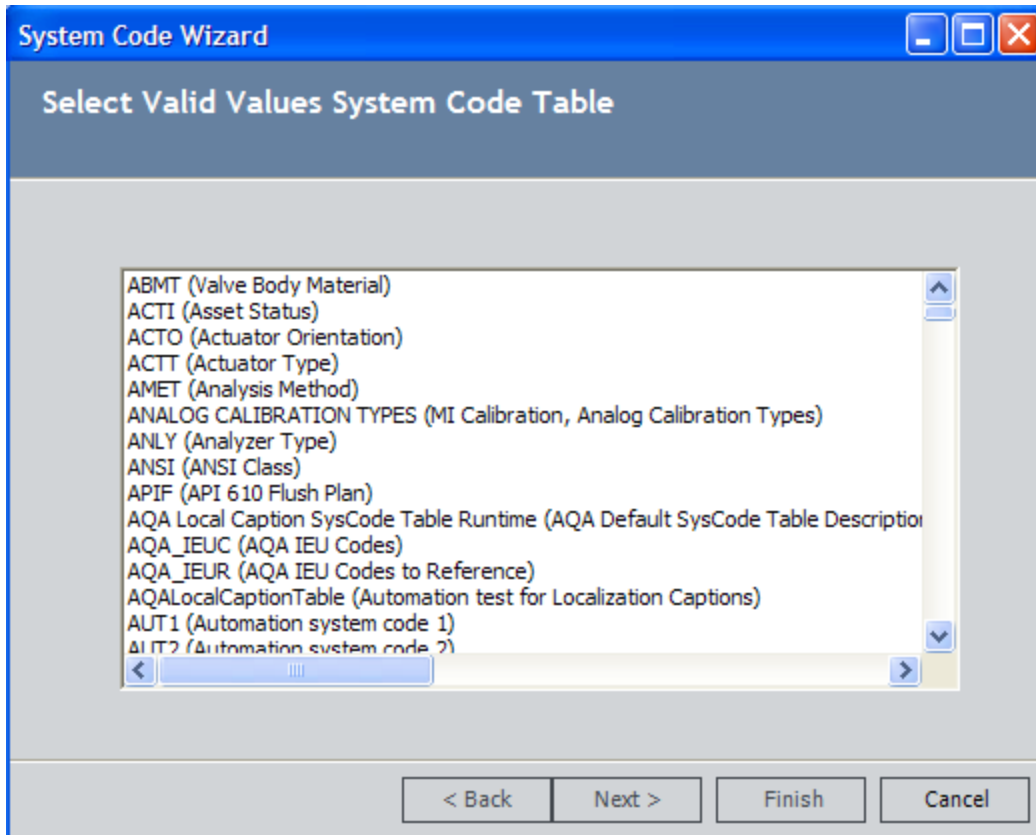
3. In the list of System Code Tables, select the System Code Table from which the list of Valid Values will be built.

   The **Select Filter** screen appears.
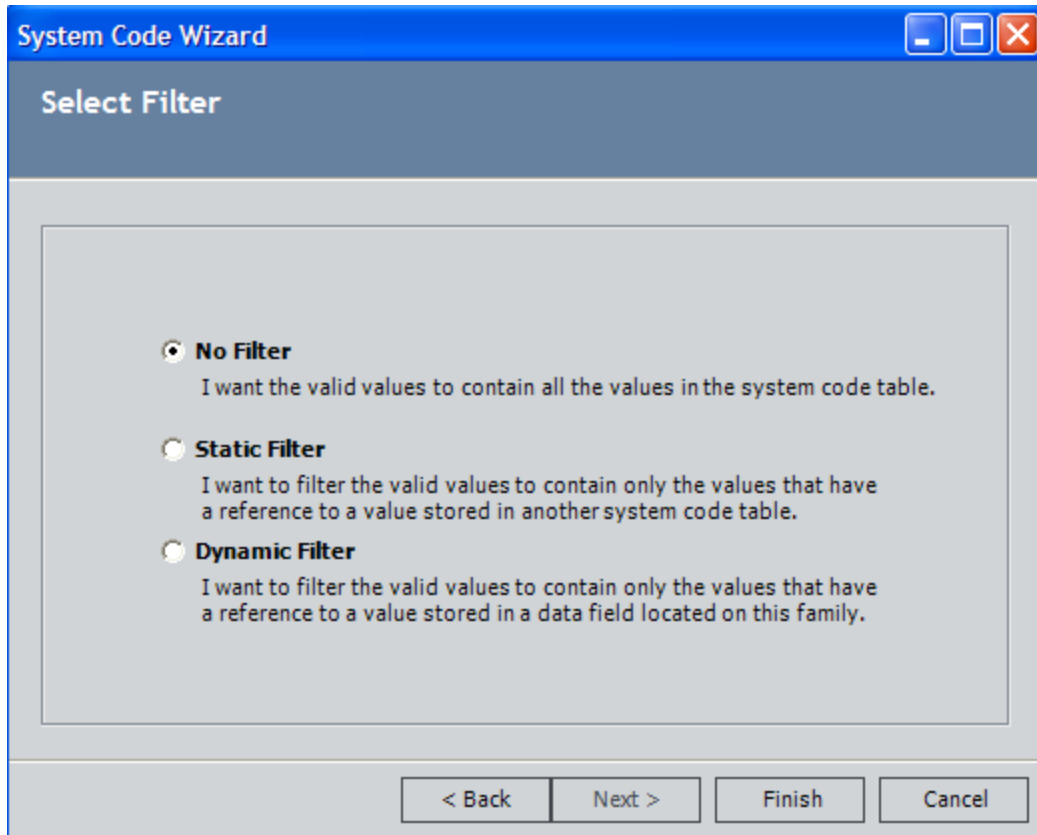
4. If you want the Valid Values list to include ALL of the System Codes in the selected System Code Table, select the **No Filter** option.

> **Note:** If the System Code Table that you selected in the previous step does not contain any references, only the **No Filter** option is available. The **Static Filter** and **Dynamic Filter** options are disabled.

-or-

If you want the Valid Values list to contain only the System Codes that are referenced from a System Code in another System Code Table...

a. Select the **Static Filter** option.

b. Click the **Next** button.

The **SelectFiltering System Code Table** screen appears.
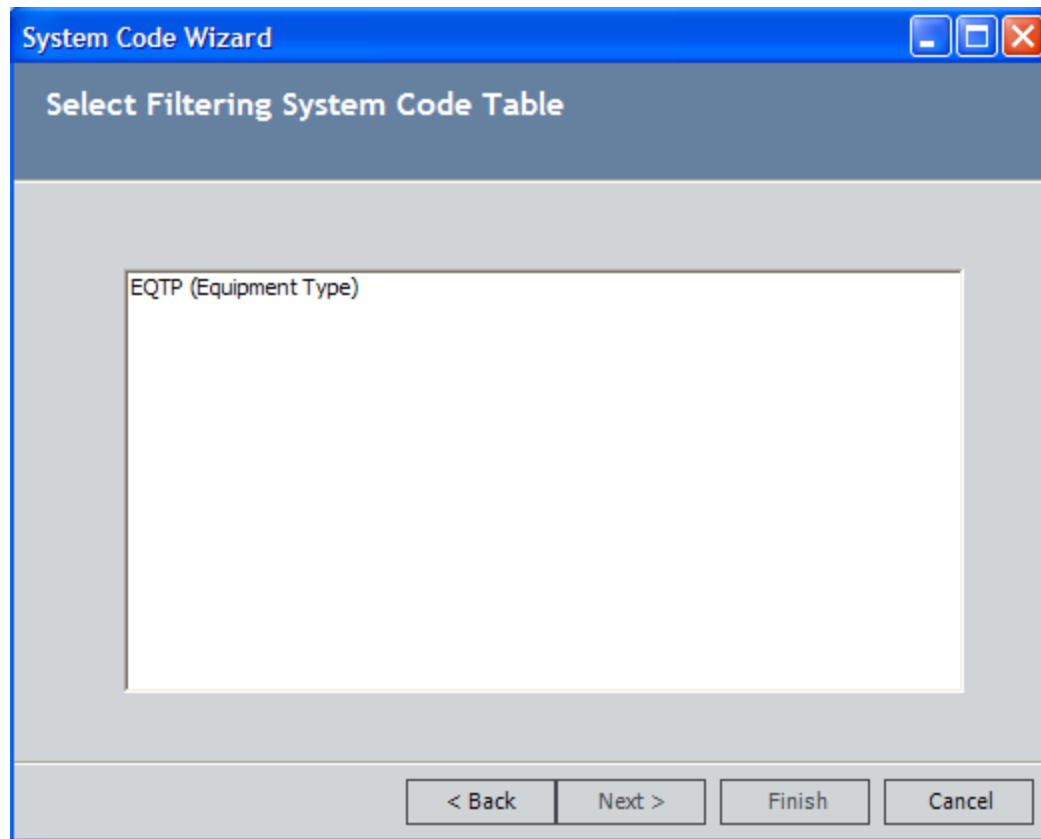
c. In the list of available System Code Tables, select the System Code Table that will be used as the filter. You should select the System Code Table that contains the System Code that references System Codes belonging to the System Code Table that you selected on the previous screen.

d. Click the **Next** button.

    The **Select Filter Value** screen appears.
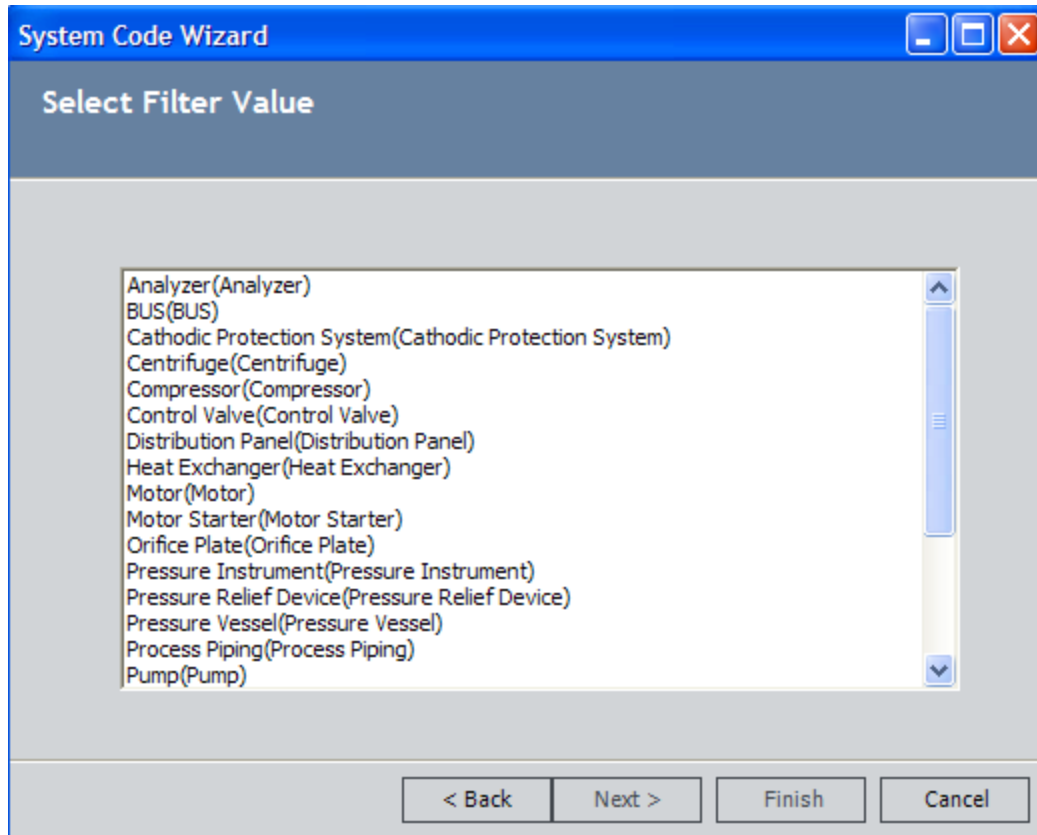
e. In the list of System Codes, select the System Code that will be used to filter the list of values in the Valid Values list. The list will contain only the System Codes from the source System Code Table that are referenced by the System Code that you select here.

f. Click the **Finish** button to close the System Code Wizard.

Click here to see an example of this functionality.

-or

If you want the Valid Values list to contain only the System Codes referenced by a System Code identified by a value stored in another field within the same record...

a. Select the **Dynamic Filter** option.

b. Click the **Next** button.

The **Select Filtering System Code Table** screen appears.
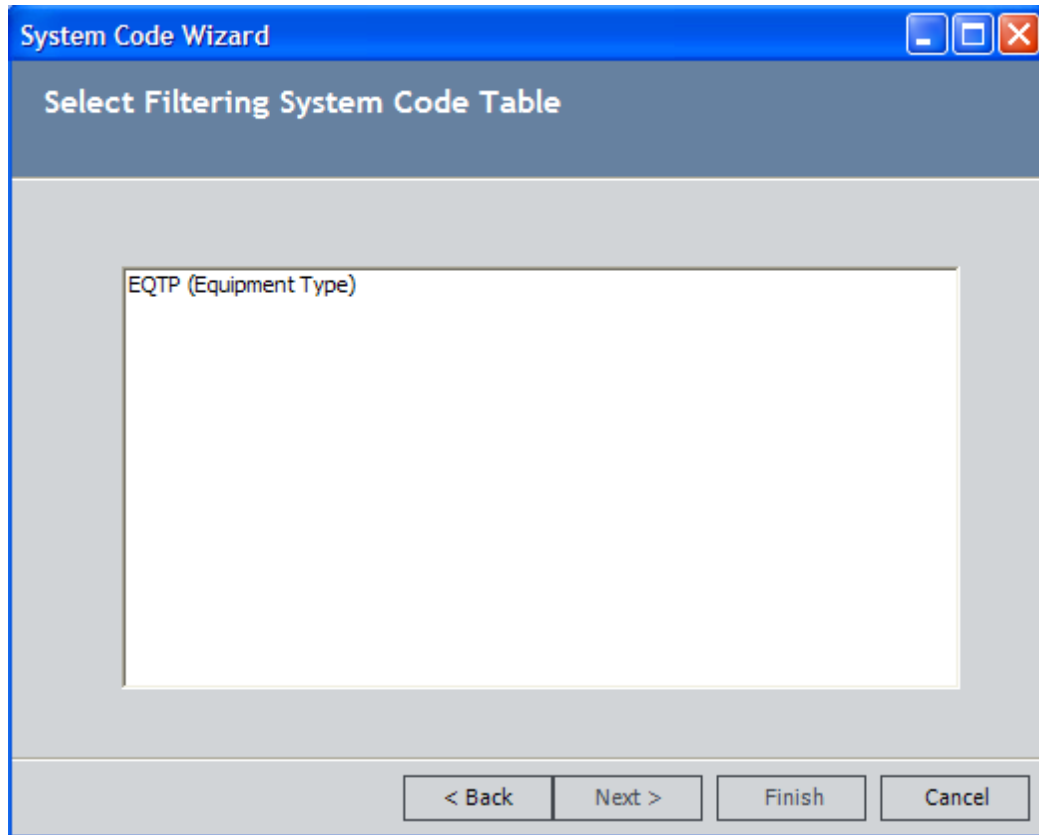
c. In the list of available System Code Tables, select the System Code Table that will be used as the filter. You should select the System Code Table that contains System Codes that reference the System Code Table that you selected on the previous screen.

d. Click the **Next** button.

The **Select Filter Value** screen appears.
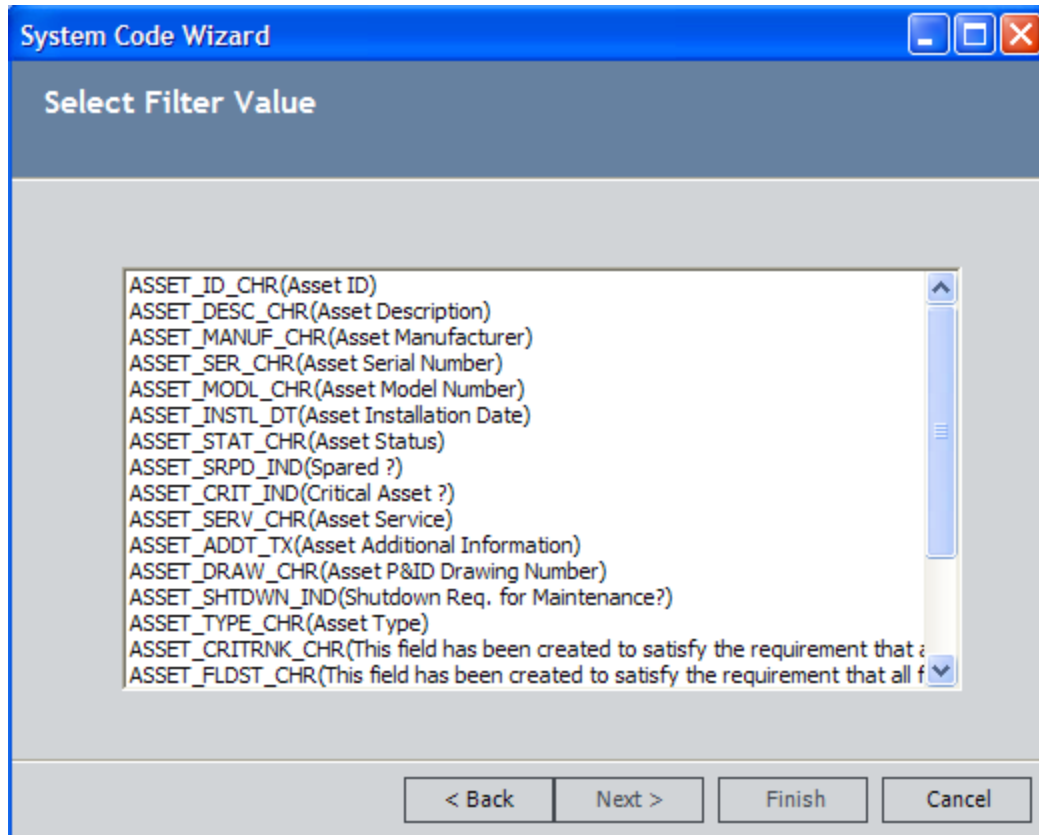
e. In the list of family fields, select the field whose value will be used to filter the list of values in the Valid Values list. The list will contain only the System Codes referenced from the System Code identified by the value in the field that you select here.

f. Click the **Finish** button to close the System Code Wizard.

   Click here to see an example of this functionality.

5. Click the **Finish** button.

   The **Rules Wizard** closes. The rule-generation process is complete.

Generating Default Value Rules

Default Value rules define the default value that will be provided for a given field. When a user creates a new record in the Record Manager, the default value will be provided automatically. Users can accept the default value or specify a different value.

To generate a Default Value rule for a field:

1. In the Configuration Manager, on the <u>View/Edit Field Properties window</u>, in the **Field Rules** section, click the **Rules Wizard** link.

   The **Rules Wizard** appears, displaying the **Rule Details** screen.

2. In the **Default Value** list, select one of the following options:

   - **Meridium APM Default:** The <u>default Meridium APM Default rule</u> for the field.

   > ⓘ **Note:** This option will be available only if a custom Default rule already exists. You can use this option to revert back to the Meridium APM Default rule.

   - **Literal Value:** The default value will be the value that you type in the text box that appears when you select this option.

      Note the following when using the Literal Value option:

   - If you type a value that you want to surround in quotation marks, you must escape the quotation marks. For example, to specify the default value **"Default"** you would need to type: `""Default""`

   - If you are defining a default value for a logical field, type `True` or `False`. For other types of fields, note that no character-length validation is performed on the value that you specify. Be sure that the value you define does not exceed the character limit of the field.

   - If you are defining a default rule for a date field, you can type `DateTime.Now` to set the default value to the current date and time (i.e., the date and time when a new record is created).

   > **Note:** The **Custom** option will also appear in the **Default Value** list if a Default rule already exists for the field, whether that rule was generated via the Rules Wizard or defined manually via the <u>family rule project</u>. Choosing the **Custom** option and proceeding through the Rules Wizard will cause the existing field-level rule to remain in tact and not be modified. Selecting any other option, however, will cause the existing field-level rule to be overwritten by the new rule code that is generated as a result of your selection.

3. Click the **Next** button.

   The **Summary** screen appears, displaying a summary of the rule that will be generated for the field.

4. Click the **Next** button.

   The **Generating Rules** screen appears, and the field-level rules are generated. A progress bar appears to indicate the status of the rule-generation process. After the field-level rules have been generated, the family rule project will be compiled automatically. A message above the progress bar on the **Generating Rules** screen will indicate the status of the compilation. After the rules have been generated and compiled, the text **Done** will appear above the progress bar, and the **Finish** button will become enabled.

5. Click the **Finish** button.

   The **Rules Wizard** closes. The rule-generation process is complete.

Generating Disabled Rules

Disabled rules determine when, if ever, the field will be disabled, or locked from editing. Disabled rules determine whether or not users can modify the value in the field.

> ⓘ **Hint:** Via the Configuration Manager, you can choose the background color that will appear on datasheets for disabled fields. The background color will be applied to all fields where rules exist to make the field disabled.

To generate a Disabled rule:

1. In the Configuration Manager, on the **View/Edit Field Properties window**, in the **Field Rules** section, click the **Rules Wizard** link.

    The **Rules Wizard** appears, displaying the **Rule Details** screen.

2. In the **Disabled** list, select one of the following options:

    - **Meridium APM Default:** The default Meridium APM Disabled rule for the field.

        > ⓘ **Note:** This option will be available only if a custom Disabled rule already exists. You can use this option to revert back to the Meridium APM Default rule.

    - **Always Disabled:** The field will always be disabled; users will never be able to modify it. You may want to disable a field, for example, if you supply a default value that should always be used or if the field value is system-generated and should not be modified.

- **Never Disabled:** The field will never be disabled; users will always be able to modify its value. If you create a new field, it will not be disabled unless you set the Disabled rule to Always Disable. The Never Disable option is useful for removing the Always Disable condition from a field. It is not necessary to set new fields explicitly to Never Disable.

> ℹ **Note:** The **Custom** option will also appear in the **Disabled** list if a Disabled rule already exists for the field, whether that rule was generated via the Rules Wizard or defined manually via the [family rule project](). Choosing the **Custom** option and proceeding through the Rules Wizard will cause the existing field-level rule to remain in tact and not be modified. Selecting any other option, however, will cause the existing field-level rule to be overwritten by the new rule code that is generated as a result of your selection.

3. Click the **Next** button.

   The **Summary** screen appears, displaying a summary of the rule that will be generated for the field.

4. Click the **Next** button.

   The **Generating Rules** screen appears, and the field-level rules are generated. A progress bar appears to indicate the status of the rule-generation process. After the field-level rules have been generated, the family rule project will be compiled automatically. A message above the progress bar on the **Generating Rules** screen will indicate the status of the compilation. After the rules have been generated and compiled, the text **Done** will appear above the progress bar, and the **Finish** button will become enabled.

5. Click the **Finish** button.

   The **Rules Wizard** closes. The rule-generation process is complete.

Generating Format Rules

Format rules determine the formatting that will be applied to values entered into a given field. For example, if a field contains a monetary value, you might define a Format rule so that if the stored value is 50, it is formatted for display as $50.00. Format rules will be applied wherever fields are displayed in *formatted* mode.

When you define a Format rule for a numeric or date field, you can choose from a list of predefined format types that will be applied to the field when it appears in the Meridium APM Framework application or the Meridium APM Web Framework. If you prefer to define a rule that is not available through one of the predefined format types, you will need to create custom rules.

To generate a Format rule for a field:

1. In the Configuration Manager, on the <u>View/Edit Field Properties window</u>, in the **Field Rules** section, click the **Rules Wizard** link.

   The **Rules Wizard** appears, displaying the **Rule Details** screen.

2. In the **Format** list, select one of the following options:

   - **Meridium APM Default:** The <u>default Meridium APM Format rule</u> for the field.

   > ⓘ **Note:** This option will be available only if a custom Format rule already exists. You can use this option to revert back to the Meridium APM Default rule.

   - If you are working with a numeric field...

- **Currency:** Specifies that the value in the field will be formatted as currency, according to the Windows Regional Options of the machine on which the Meridium APM Framework application is running and the number of digits past the decimal that you specify in the Configuration Manager. For currency fields, you will likely want to choose this option only if all the values in the database are the same currency and all the machines in your organization have the same Regional Options settings. When you select this option, the example that appears at the bottom of the **Format** dialog box uses the currency symbol associated with the Regional Options setting from the machine on which you are running the Configuration Manager application.

> ⓘ **Note:** If you need to format a field to display the appropriate currency depending on a user's culture setting or some other criteria, you will need to define a custom rule.

- **Fixed Digits Before Decimal Point:** Specifies the number of digits that will be displayed before the decimal point.
- **Scientific:** Specifies that field values will be formatted as a scientific number.
- **Fixed Digits After Decimal Point:** Specifies the number of digits that will appear to the right of the decimal point.
- **General(Scientific or Fixed, whichever is shorter):** Specifies that the value will be displayed in fixed or scientific format, depending on which one is shorter.
- **Formatted Number(with thousands separator):** Specifies that the thousands separator will be displayed. Additionally, with this option, you can specify the number of digits that will appear to the right of the decimal.
- **Hexadecimal:** Specifies that field values will be formatted as hexadecimal numbers. Additionally, with this option, you can specify the minimum number of digits so that leading zeros will be added to values with fewer than the specified number of digits.

> **Note:** Setting the Format rule to **Hexadecimal** will cause the Meridium APM system to generate an error when a value is entered into the field via a datasheet in the Meridium APM Framework. We recommend that you do not use this option.

- **Percent:** Specifies that the field will be formatted as a percentage, including a percent sign (%) and the number of digits that you specify to the right of the decimal.
- **Custom:** Specifies that field values will be formatted according to the custom options that you define. With a Custom format, you can define one or more characters to appear either to the right or left of the value. Additionally, you can define the number of digits that will appear to the right of the decimal by typing .0000, where each zero that you type indicates a decimal place that will be displayed. You can type as many zeros as you want.

- **If you are working with a date field...**
  - **Short Date:** For example, 3/31/2005.
  - **Long Date:** For example, Thursday, March 31, 2005.
  - **Short Time:** For example, 4:27 PM.
  - **Long Time:** For example, 4:27:00 PM.
  - **Date and Time Short:** For example, Thursday, March 31, 2005, 4:27 PM.
  - **Date and Time Long:** For example, Thursday, March 31, 2005, 4:27:00 PM.
  - **General Short Time:** For example, 3/31/2005 4:27 PM.
  - **General Long Time:** For example, 3/31/2005 4:27:00 PM.
  - **RFC 1123:** For example, Thu, 31 March 2005 16:27:00 GMT.
  - **ISO 8601:** For example, 2005-03-31T16:27:00.
  - **Month Day:**For example, March 31.
  - **Year Month:** For example, March, 2005.
  - **Custom Format:** The format that you specify in the text box. Use the following characters to specify each digit in the format. Note that the characters are case sensitive.
    - **M:** Month.

> **Note:** The **Custom** option will also appear in the **Format** list if a Format rule already exists for the field, whether that rule was generated via the Rules Wizard or defined manually via the [family rule project](#). Choosing the **Custom** option and proceeding through the Rules Wizard will cause the existing field-level rule to remain in tact and not be modified. Selecting any other option, however, will cause the existing field-level rule to be overwritten by the new rule code that is generated as a result of your selection.

3. Click the **Next** button.

   The **Summary** screen appears, displaying a summary of the rule that will be generated for the field.

4. Click the **Next** button.

   The **Generating Rules** screen appears, and the field-level rules are generated. A progress bar appears to indicate the status of the rule-generation process. After the field-level rules have been generated, the family rule project will be compiled automatically. A message above the progress bar on the **Generating Rules** screen will indicate the status of the compilation. After the rules have been generated and compiled, the text **Done** will appear above the progress bar, and the **Finish** button will become enabled.

5. Click the **Finish** button.

   The **Rules Wizard** closes. The rule-generation process is complete.

Generating Rules for Multiple Fields

When you access the Rules Wizard from the **View/Edit Field Properties** window, you can generate field-level rules for a single field. You also have the option of accessing the Rules Wizard from the **Manage Family Fields** window to generate field-level rules for more than one field at once.

To generate field-level rules for multiple fields:

1. In the Configuration Manager, on the **Manage Family Fields** window, click the **Rules Wizard** link.

   The **Rules Wizard** appears, displaying the **Select Fields** screen.

   The **Select Fields** screen displays a list of all the fields that are defined within the family that was displayed on the **Manage Family Fields** window. The list contains fields that are defined within that family itself and fields that are spread from a family but for which you are allowed to modify the properties at the subfamily level.

2. In the **ID** column of the fields for which you want to generate rules, select the check box.

   > ⓘ **Hint:** You can click the **Select All** link to select all the fields in the list. You can click the **Clear All** link to clear you current selections.

3. Click the **Next** button.

   The **Rule Details** screen appears.

4. Use the available options to define the rules that you want to generate. Any option that is not supported for any of the fields that you selected will be disabled. You can use the instructions in any of the following topics for details on configuring specific types of rules.

   - Generating Required Rules
   - Generating Validation Rules
   - Generating Valid Values Rules
   - Generating Default Value Rules
   - Generating Disabled Rules
   - Generating Format Rules

5. Click the **Next** button.

   The **Summary** screen appears, displaying a summary of the rule that will be generated for the field.

6. Click the **Next** button.

The **Generating Rules** screen appears, and the field-level rules are generated. A progress bar appears to indicate the status of the rule-generation process. After the field-level rules have been generated, the family rule project will be compiled automatically. A message above the progress bar on the **Generating Rules** screen will indicate the status of the compilation. After the rules have been generated and compiled, the text **Done** will appear above the progress bar, and the **Finish** button will become enabled.

7. Click the **Finish** button.

The **Rules Wizard** closes. The rule-generation process is complete.

How to Create Custom Field-Level Rules

You can develop custom field-level rules manually by [accessing the code item for the desired field](#) and inserting the desired custom code. The code for each rule type (Required, Validation, Valid Values, Default Value, Disabled, and Format) must be defined within the appropriate function, which serves as a container for the code for that rule. These functions are inserted automatically when you generate field-level rules via the Rules Wizard, so if you need to modify existing rules, the functions should already exist. If you want to create custom rules from scratch, however, you will need to insert the appropriate function(s) into the field class and then develop the desired rule code within those functions.

For more information on developing custom rules, see the [examples provided of rules that invoke specific behaviors](#).

About Functions Associated with Field Rule Types

Each type of field-level rule has a function associated with it. When you generate rules using the Rules Wizard, these functions and the code that is necessary to build the rule that you chose to define are inserted into the code item for the selected field(s). The content of the functions will vary, depending upon the rules that you create in the Rules Wizard.

In this area of the documentation, we provide a code excerpt that shows an example of the function behind each rule type. Each example in this section can be built using the Rules Wizard. The examples are not meant to take the place of the Rules Wizard and are not intended to be copied and pasted into your rules. Rather, they serve as a reference point that may allow you to better understand and interpret each part of a field's code item if you view it in the Meridium APM Rules Editor.

If you want to develop a rule that is more complex than those supported by the Rules Wizard, you can use the examples as a starting point for inserting custom code into your rule projects. Alternatively, you can use the Rules Wizard to develop a simple rule and then modify the rule by editing the rule project directly.

Required

The IsRequired function allows you to create a rule that will make a field *required*, meaning that users must enter a value into the field before they can save a record in that family. The following code excerpt provides an example of the Required function. You can insert this code into the code item for a field to make that field required.

```
-----------------------------------------------------------------------

Public Overrides Function IsRequired() As Boolean

        Return True

End Function

-----------------------------------------------------------------------
```

Note that:

- Changing *True* to *False* in this example will make the field not required.
- If this function does not exist in the code item for a field, that field will be *optional*.

Validation

The Validate function allows you to create a Validation rule for a field. Validation rules allow you to develop logic that will be executed when users enter values into a field to ensure that the value that has been entered is valid. Validation logic can range from very simple to highly complex. The following code excerpt provides an example of a Validation function. This example validates that the value entered into a field is greater than 100.

```
-------------------------------------------------------------------------

Public Overrides Function Validate(ByRef newValue As Object) As FieldValidationStatus

        If Not Convert.IsDBNull(newValue) Then

            If CDbl(newValue) > CDbl("100") Then

                    Return FieldValidationStatus.Success()

                Else

                    Return FieldValidationStatus.Failure()

                End If


        Else

            Return FieldValidationStatus.Success()

        End If

End Function

-------------------------------------------------------------------------
```

This Validation rule shows a basic example of validation code that consists of two *If...Else...* statements, one shown in blue and the other shown in red.

- The first If...Else... statement (in blue) checks whether or not the field contains a value. If the field is *empty*, validation is considered successful. No action is taken, and no feedback is provided to the user.
- If the field contains a value, the second If...Else... statement (in red) is executed. In this portion of the code...
- The line *If CDbl(newValue) > CDbl("100") Then* defines the validation condition. In this example, the rule is validating that *newValue* (i.e., the value entered into the field) is greater than 100. The content of this code will vary, depending upon the type of field and the validation that is being applied.
- If the field meets the validation condition, validation is successful, and no feedback is provided to the user.
- If the field does not meet the validation condition, validation fails, and a message is displayed to the user indicating that the value is invalid.

Valid Values

The GetPickList function allows you to create a Valid Values rule for a field. The following code excerpt provides an example of a Valid Values rule that would cause a field to be populated with the values 1 and 2.

```
--------------------------------------------------------------------

Public Overrides Function GetPickList() As DynamicPickList

        Dim pickList As DynamicPickList

        pickList = MyBase.CreatePickList(True)

        PopulatePickList(pickList)

        Return pickList

End Function

Public Overrides Sub PopulatePickList(ByVal pickList As DynamicPickList)

        pickList.AddRange("1|2".Split("|"c))

End Sub

--------------------------------------------------------------------
```

> 🛈 **Note:** The GetPicklist function simply references the PopulatePicklist procedure. For Valid Values rules, the code that defines the behavior of the field will reside within the PopulatePicklist procedure.

Default Value

The GetDefaultInitialValue function allows you to create a Default Value rule for a field. The following code excerpt provides an example of a Default Value rule that would set the value of a field to 1 by default.

```
-----------------------------------------------------------------------

Public Overrides Function GetDefaultInitialValue() As Object

        Return CDbl("1")

End Function

-----------------------------------------------------------------------
```

The line *Return CDbl("1")* specifies the default value to use. The content of this line of code will vary, depending upon the field type and the default value itself.

Disabled

The IsDisabled function allows you to develop a Disabled rule for a field. Disabled rules determine whether or not the value in a field can be modified. The following code excerpt provides an example of a Disabled rule that will cause a field to be disabled, meaning that it is read-only and not editable.

Public Overrides Function IsDisabled() As Boolean

    Return True

End Function

Note that:

- Changing *True* to *False* in this example will make the field not required.
- If this function does not exist in the code item for a field, that field will be *optional*.

Format

The FormatValue function allows you to create Format rules for a field. Format rules determine how the value stored in a field will be modified for display purposes. The following code excerpt provides an example of a Format rule that will cause numeric values to be displayed with two decimal places.

Public Overrides Function FormatValue(ByVal value As Object) As String

>   If value Is DBNull.Value Then Return String.Empty Else Return CDbl(value).ToString ("f2")

End Function

You can also use the FormatValue function to format the background color of the datasheet cell for a particular field. The following code excerpt provides an example of a Format rule that will cause the datasheet cell to be colored *GreenYellow*.

Public Overrides Function FormatValue(ByVal value As Object) As String

>   MyBase.CurrentDataField.BackColor = Drawing.Color.GreenYellow

>   Return MyBase.FormatValue(value)

End Function

As a result of using this rule, you can see in the following image that the datasheet cell is colored *GreenYellow*.
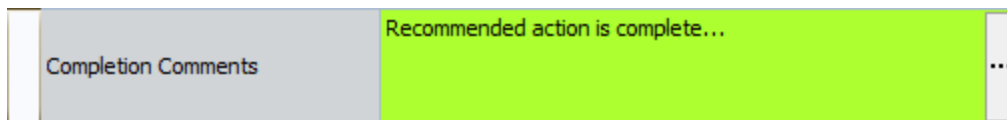
Note also that if a field is formatted in this way and you include it in a query that is running in *formatted* mode, the query results will also show the formatted color, as shown in the following image.

About These Examples

In this area of the documentation, we provide examples of field-level rules that you can use to invoke specific behaviors. While these examples are not comprehensive and do not describe all the behaviors that you can implement through rules, they should serve as a starting point for allowing you to understand the flexibility that can be achieved through business rule and help you get started writing your own rule code. You can also use these examples as a way to better understand the rules that already exist in your rule code.

In most cases, we limit these examples to rule that cannot be constructed by using the Rules Wizard. Most of the rules that can be constructed through the Rules Wizard are fairly simple and self-explanatory. To see examples of these rules, use the Rules Wizard to select different options and the view the result in the Meridium APM Framework application.

Making a Field Required Based on the Value in Another Field

Instead of making a field *always* required, you may prefer to make it required only if another field in the same record also contains a value. For example, assume that you have a Work Order family that contains the logical field *Approved* (EWORK_APPRO_FLG), which is meant to be flagged by a manager after a Work Order record has been reviewed and approved. The Work Order family also contains the *Approved By* field, which is meant to contain the name of the manager who made the approval. Therefore, you want to create a rule so that when the Approved field is set to *True*, the Approved By field is required.

To enforce this condition, you would create a Required rule on the Approved By field that looks like this:

Public Overrides Function IsRequired() As Boolean

     If Object.Equals (CurrentEntity.Fields("EWORK_APPRO_FLG").Value, True) Then

          Return True

     Else

          Return False

     End If

End Function

The portion of the code shown in blue (EWORK_APPRO_FLG) identifies the Field ID of the field that you want to use for enforcing the Required rule condition, and the portion shown in green (True) specifies the value that the field must contain.

Displaying a Custom Message When a Field Fails Validation

If a Validation rule exists for a field, when a user enters an invalid value into that field, the Meridium APM system will display a standard error message. For example, assume that the Work Order family contains the Repair Time field, which is required and must contain a value that is greater than zero. In this case, you could use the Rules Wizard to generate a Validation rule that will ensure that the value entered into the field is greater than zero. The rule that is generated will use the standard error message, so when user enters an invalid value, the following error message will appear.
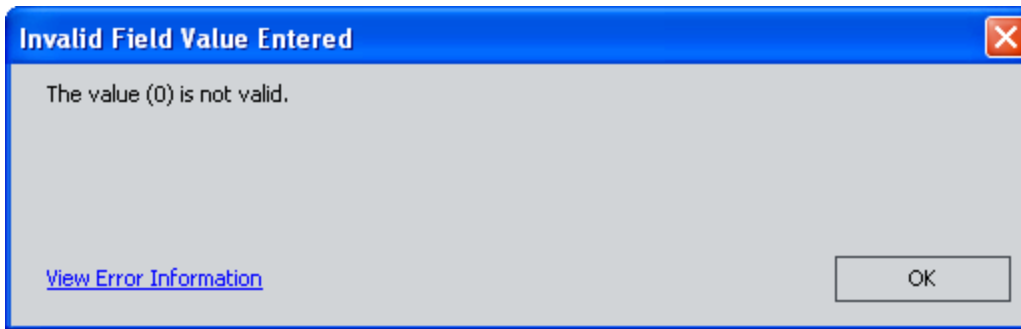
If instead you prefer to display a message that is more descriptive and provides the user with additional guidance on what value to enter, you could modify the default message. To do so, you will need to edit the rule code.

The rule code that is generated by the Rules Wizard for the rule in this example looks like this:

```
------------------------------------------------------------------------

Public Overrides Function Validate(ByRef newValue As Object) As FieldValidationStatus
        If Not Convert.IsDBNull(newValue) Then
                If CDbl(newValue) > CDbl("0") Then
                Return FieldValidationStatus.Success()
        Else
                Return FieldValidationStatus.Failure()
        End If
        Else
                Return FieldValidationStatus.Success()
        End If
End Function


------------------------------------------------------------------------
```

If you wanted to modify the rule to display the message, "You must enter a value greater than 0 (zero)," you would modify rule to look like this:

```
------------------------------------------------------------------------

Public Overrides Function Validate(ByRef newValue As Object) As FieldValidationStatus
```

```
        If Not Convert.IsDBNull(newValue) Then
                If CDbl(newValue) > CDbl("0") Then
                Return FieldValidationStatus.Success()
        Else
                Return FieldValidationStatus.Failure("You must enter a value greater than 0
                (zero).")
        End If
        Else
                Return FieldValidationStatus.Success()
        End If
End Function
```

Notice that the changes to the rule are shown in blue. You can replace the blue text with the text of your choice to display the desired message.

• Page 214 of 529

Validating the Value in One Field Against the Value in Another

For some fields, it may be useful to develop Validation rules that take into account the values in other fields in the same record. Consider an example where the Work Order family contains the fields Maintenance Type (EWORK_MAINT_TYPE_CHR) and Labor Cost (EWORK_LABOR_COST_NBR). Since the labor cost will vary depending upon the type of work that is represented by the work order, you may want to develop rules that validate the value entered in the Labor Cost field against the value selected in the Maintenance Type field. For instance, assume that two maintenance types are supported: *Repair* and *Clean*. If the Maintenance Type is *Repair*, then it is assumed that there is *some* cost associated with the repair, so the Labor Cost field must be greater than zero. Similarly, you might want to specify a condition whereby if the Maintenance Type is *Clean*, the Labor Cost cannot exceed $500.00.

You could enforce both of these scenarios by creating the following Validation rule for the Labor Cost field:

```
--------------------------------------------------------------------------

Public Overrides Function Validate(ByRef newValue As Object) As FieldValidationStatus
        If Not Convert.IsDBNull(newValue) Then
                If Not Convert.IsDBNull(CurrentEntity.Fields("EWORK_MAINT_TYPE_
                CHR").Value) Then
                        Select Case CStr(CurrentEntity.Fields("EWORK_MAINT_TYPE_
                        CHR").Value)
                                Case "Repair"
                                        If Convert.ToInt32(newValue) > 0
                                                Return FieldValidationStatus.Success()
                                        Else
                                                Return FieldValidationStatus.Failure("The Labor Cost
                                                for a repair must be greater than 0 (zero).")
                                        End If
                                Case "Clean"
                                        If Convert.ToInt32(newValue) >= 0 And Convert.ToInt32
                                        (newValue) <= 500 Then
                                                Return FieldValidationStatus.Success()
                                        Else
                                                Return FieldValidationStatus.Failure("The Labor Cost
                                                for cleaning must be less than $500.")
                                        End If
                                Case Else
                                        Return FieldValidationStatus.Failure(CStr(Cur-
                                        rentEntity.Fields("EREPAI_REPAI_TYPE_CHR").Value) &
                                        " is not a valid Repair Type.")
                        End Select
                Else
                        Return FieldValidationStatus.Failure("You must first select a
                        Repair Type.")
                End If
```

```
        Else
                Return FieldValidationStatus.Success()
    End If
End Function
```

By applying this Validation rule to the Labor Cost field:

- If a user selects *Repair* in the Maintenance Type field and then enters a value of 0 (zero) in the Labor Cost field, the following message will be displayed: *The Labor Cost for a repair must be greater than 0 (zero)*. The code that defines this behavior is:

    ```
    Case "Repair"
            If Convert.ToInt32(newValue) > 0
                    Return FieldValidationStatus.Success()
            Else
                    Return FieldValidationStatus.Failure("The Labor Cost for a repair must
                    be greater than 0 (zero).")
            End If
    ```

- If a user selects *Clean* in the Maintenance Type field and then enters a value that is greater than 500 in the Labor Cost field, the following message will be displayed: *The Labor Cost for cleaning must be less than $500.* The code that defines this behavior is:

    ```
    Case "Clean"
            If Convert.ToInt32(newValue) >= 0 And Convert.ToInt32(newValue) <=
            500 Then
                    Return FieldValidationStatus.Success()
            Else
                    Return FieldValidationStatus.Failure("The Labor Cost for cleaning
                    must be less than $500.")
            End If
    ```

> ⓘ **Hint:** You could repeat the code between *Case"Value"* and *End If* as many times as needed, depending upon how many values that you wanted to validate against. You would simply modify the value, the validation condition, and the message as needed.

- If a user types a value other than *Repair* or *Clean* in the Maintenance Type field, when the user enters ANY value in the Labor Cost field, the following message will be displayed: *[Entered Value] is not a valid Repair Type.* [Entered Value] will be replaced with the value that the user entered. The code that defines this behavior is:

```
-----------------------------------------------------------------------

Case Else
            Return FieldValidationStatus.Failure(CStr(CurrentEntity.Fields("EREPAI_
            REPAI_TYPE_CHR").Value) & " is not a valid Repair Type.")
End Select

-----------------------------------------------------------------------
```

- If a user enters a value in the Labor Cost field without first selecting a value in the Maintenance Type field, the following message will be displayed: *You must first select a Repair Type.* The code that defines this behavior is:

```
-----------------------------------------------------------------------

Else
        Return FieldValidationStatus.Failure("You must first select a Repair Type.")
End If

-----------------------------------------------------------------------
```

Populating a Field Automatically Based on the Value in Another Field

You can use Validation rules to conditionally populate other fields based upon the value in the field for which the Validation rule exists. Consider an example of the Motor family, which contains two fields:

- **In Service (AEMOT_IN_SERVI_FLG):** A logical field that specifies whether or not the motor is currently being used.
- **In Service Date (AEMOT_IN_SERVI_DATE_DT):** A date field that stores the date on which the motor was put in service.

The In Service Date field needs to be populated only for motors that are actually in service. So you would not want to create a Default Value rule to populate the field automatically in ALL Motor records. Instead, you would want to create a Validation rule that will populate the field automatically when the In Service field is set to *True*. To implement this behavior, you would create the following Validation rule on the In Service field:

```
-------------------------------------------------------------------

Public Overrides Function Validate(ByRef newValue As Object) As FieldValidationStatus

        If Not Convert.IsDBNull(newValue) Then

                If Object.Equals(newValue, True) Then

                        CurrentEntity.Fields("AEMOT_IN_SERVI_DATE_DT").Value = System.DateTime.Now

                Else

                        CurrentEntity.Fields("AEMOT_IN_SERVI_DATE_DT").Value = DBNull.Value

                End If

                Return FieldValidationStatus.Success()

        Else

                Return FieldValidationStatus.Success()

        End If

End Function

-------------------------------------------------------------------
```

In this example, the code consists of two If...Else... statements similar to the code constructed for a [basic Validation rule](). Within the second If...Else... statement, instead of the FieldValidationStatus.Success and FieldValidationStatus.Failure events, the following events are specified:

- If the In Service field is set to *True*, the In Service Date field is populated with the current date and time. This behavior is accomplished with the following code:

  ```
  ----------------------------------------------------------------------

  If Object.Equals(newValue, True) Then

          CurrentEntity.Fields("AEMOT_IN_SERVI_DATE_DT").Value = Sys-
          tem.DateTime.Now


  ----------------------------------------------------------------------
  ```

- If the In Service field contains any other value (in this case, the only other value is *False*), the In Service Date field is not populated. This behavior is accomplished with the following code:

  ```
  ----------------------------------------------------------------------

  Else

          CurrentEntity.Fields("AEMOT_IN_SERVI_DATE_DT").Value = DBNull.Value


  ----------------------------------------------------------------------
  ```

> ⓘ **Hint:** Notice that this Validation rule does not contain the line *Return FieldValidationStatus.Failure* that is included in most Validation rules. This means that there is no failure condition in this rule.

Making a Valid Values List Unrestricted

By default, a [Valid Values list created by the Rules Wizard](#) is *restricted*, meaning that users will only be allowed to select from the values that are displayed in the list. They will not be able to type their own values in the list. If you prefer, you can make the list *unrestricted*, so that users can type their own values in the list.

For example, the following code excerpt show a Valid Values rule that will construct a list containing the values **A**, **B**, and **C**.

```
------------------------------------------------------------------------

Public Overrides Function GetPickList() As DynamicPickList
        Dim pickList As DynamicPickList
        'Use MyBase.CreatePickList(True)  if you want Restricted   PickList
        'Use MyBase.CreatePickList(False) if you want Unrestricted PickList
        pickList = MyBase.CreatePickList(True)
        PopulatePickList(pickList)
        Return pickList
End Function
Public Overrides Sub PopulatePickList(ByVal pickList As DynamicPickList)
        pickList.AddRange("A|B|C".Split("|"c))
End Sub


------------------------------------------------------------------------
```

The line of code that determines that the list is restricted is **pickList = MyBase.CreatePickList(True)**. You can make the list unrestricted by changing True to False, as follows:

pickList = MyBase.CreatePickList(False)

Adding a Blank Value to a Valid Values List

By default, static <u>Valid Values lists created via the Rules Wizard</u> will not contain a blank value. A blank value can be useful for *clearing* a previously selected value. When a user selects the blank value, it will clear whatever value had previously been selected in the list. If desired, you can add a blank value to a static Valid Values list by using the following guidelines.

For example, the following code excerpt show a static Valid Values rule that will construct a list containing the values **A**, **B**, and **C**.

```
------------------------------------------------------------------------

Public Overrides Function GetPickList() As DynamicPickList
        Dim pickList As DynamicPickList
        'Use MyBase.CreatePickList(True)  if you want Restricted   PickList
        'Use MyBase.CreatePickList(False) if you want Unrestricted PickList
        pickList = MyBase.CreatePickList(True)
        PopulatePickList(pickList)
        Return pickList
End Function
Public Overrides Sub PopulatePickList(ByVal pickList As DynamicPickList)
        pickList.AddRange("A|B|C".Split("|"c))
End Sub


------------------------------------------------------------------------
```

You can add a null value to the list by inserting the code **pickList.Add(DBNull.Value)** above the line **pickList.AddRange ("A|B|C" .Split ("|"c))**. The resulting code would look like this:

```
------------------------------------------------------------------------

Public Overrides Function GetPickList() As DynamicPickList
        Dim pickList As DynamicPickList
        'Use MyBase.CreatePickList(True)  if you want Restricted   PickList
        'Use MyBase.CreatePickList(False) if you want Unrestricted PickList
        pickList = MyBase.CreatePickList(True)
        PopulatePickList(pickList)
        Return pickList
End Function
Public Overrides Sub PopulatePickList(ByVal pickList As DynamicPickList)
        pickList.Add(DBNull.Value)
        pickList.AddRange("A|B|C".Split("|"c))
End Sub


------------------------------------------------------------------------
```

Note: Valid Values lists created by the Rules Wizard to be based on System Code Tables always contain a blank value by default. If you do not want the list to contain a blank value, simply remove the code **pickList.Add(DBNull.Value)** from the rule.

Defining a Custom Sort Order for a Valid Values List

You can define a custom sort order by inserting custom logic into a Valid Values rule. For example, the Calibration rules library project and the Calibration, Analog family rule project contain logic that applies a custom sort order to the Calibration Strategy field in Calibration Event records.

The following text shows an excerpt from the Calibration *rules library* project, where the text in blue identifies the portion of the rule that creates the framework for the custom sort logic and the text in green identifies the custom sort logic itself. If you want to define custom sort logic for a field, you can insert the following text and replace the green text with your custom logic.

```
Public ClassAnalogStrategyComparer
       Implements IComparer
Public Function Compare(ByVal x As Object, ByVal y As Object) As Integer _
        Implements System.Collections.IComparer.Compare
       Dim DirectionOrder As New Hashtable
       DirectionOrder.Add("U", 1) 'Up direction
       DirectionOrder.Add("D", 2) 'Down direction
       DirectionOrder.Add("UD", 3) 'Up/Down direction
       Dim splitter() As Char = {CChar("/")}
       Dim strategyXSplit() As String = DirectCast(x, PickListItem).Value.ToString.Split(splitter)
       Dim strategyYSplit() As String = DirectCast(y, PickListItem).Value.ToString.Split(splitter)
       Dim strategyXNumber() As String = New String() {String.Empty, String.Empty}
       For Each s As String In strategyXSplit
       For Each character As Char In s
       If IsNumeric(character) Then
       strategyXNumber(Array.IndexOf(strategyXSplit, s)) &= character
       End If
       Next
       Next
       Dim strategyYNumber() As String = New String() {String.Empty, String.Empty}
       For Each s As String In strategyYSplit
       For Each character As Char In s
       If IsNumeric(character) Then
       strategyYNumber(Array.IndexOf(strategyYSplit, s)) &= character
       End If
       Next
       Next
       Dim strategyXDir As String = String.Empty
       For Each s As String In strategyXSplit
       For Each character As Char In s
       If Not IsNumeric(character) Then
       strategyXDir &= character
       End If
       Next
       Next
```

```vb
Dim strategyYDir As String = String.Empty
For Each s As String In strategyYSplit
For Each character As Char In s
If Not IsNumeric(character) Then
strategyYDir &= character
End If
Next
Next
If (Not DirectionOrder.Contains(strategyXDir) And Not DirectionOrder.Contains
(strategyYDir)) Then
Return String.Compare(DirectCast(x, PickListItem).DisplayValue, DirectCast(y, Pick-
ListItem).DisplayValue)
ElseIf (Not DirectionOrder.Contains(strategyXDir) And DirectionOrder.Contains
(strategyYDir)) Then
Return 1
ElseIf (DirectionOrder.Contains(strategyXDir) And Not DirectionOrder.Contains
(strategyYDir)) Then
Return -1
End If
If strategyXDir = "UD" And strategyXNumber(1) = String.Empty Then
strategyXNumber(1) = strategyXNumber(0)
End If
If strategyYDir = "UD" And strategyYNumber(1) = String.Empty Then
strategyYNumber(1) = strategyYNumber(0)
End If
If (Not IsNumeric(strategyXNumber(0)) And Not IsNumeric(strategyYNumber(0)))
Then
Return String.Compare(DirectCast(x, PickListItem).DisplayValue, DirectCast(y, Pick-
ListItem).DisplayValue)
ElseIf (Not IsNumeric(strategyXNumber(0)) And IsNumeric(strategyYNumber(0)))
Then
Return 1
End If
If (CInt(strategyXNumber(0)) < CInt(strategyYNumber(0))) Then
Return -1
ElseIf (CInt(strategyXNumber(0)) = CInt(strategyYNumber(0))) Then
If (CInt(DirectionOrder(strategyXDir)) < CInt(DirectionOrder(strategyYDir))) Then
Return -1
ElseIf (CInt(DirectionOrder(strategyXDir)) = CInt(DirectionOrder(strategyYDir))) Then
If (IsNumeric(strategyXNumber(1).ToString) And IsNumeric(strategyYNumber
(1).ToString)) Then
If (CInt(strategyXNumber(1)) < CInt(strategyYNumber(1))) Then
Return -1
ElseIf (CInt(strategyXNumber(1)) = CInt(strategyYNumber(1))) Then
Return 0
ElseIf (CInt(strategyXNumber(1)) > CInt(strategyYNumber(1))) Then
Return 1
End If
Else
Return 0
End If
ElseIf (CInt(DirectionOrder(strategyXDir)) > CInt(DirectionOrder(strategyYDir))) Then
```

```
            Return 1
            End If
        ElseIf (CInt(strategyXNumber(0)) > CInt(strategyYNumber(0))) Then
            Return 1
            End If
            Return 0
    End Function
End Class
```

> ⓘ **Note:** Commented text has been removed from the following code snippet.

The following text shows an excerpt from the Calibration, Analog *family rule project*, where the text in blue identifies the portion of the rule that creates the framework for the custom sort logic and the text in green identifies the custom sort logic itself. If you want to define custom sort logic for a field, you can insert the following text and replace the green text with your custom logic.

```
Public Overrides Function GetPickList() As DynamicPickList
        Dim pickList As DynamicPickList
        pickList = MyBase.CreatePickList(True)
        If CurrentEntity.Family.ID = "MI_EVCADSCT" Then
        pickList.AddSystemCodes("MI_CALIBRATION_STRATEGIES", "MI_CALIBRATION_
        REFERENCES", "CALIBRATION DISCRETE STRATEGIES")
        Return pickList
        Else
        pickList.AddSystemCodes("MI_CALIBRATION_STRATEGIES", "MI_CALIBRATION_
        REFERENCES", "CALIBRATION ANALOG STRATEGIES")
        Dim sortedPicklist As PickList = pickList.Sort(New AnalogStrategyComparer())
        pickList.Clear()
        pickList.AddRange(sortedPicklist)
        Return pickList
        End If
End Function
```

> ⓘ **Note:** Commented text has been removed from the following code snippet.

Filtering a System Code List to Display Referenced System Codes

By using System Code references, you can develop a Valid Values rule to display only the System Codes that are referenced from another System Code. Consider, for example, two System Code Tables: Manufacturer (MFGR) and Equipment Type (EQPT).

- The Equipment Type table contains the following values:
  - Compressor
  - Heat Exchanger
  - Motor
  - Pump
  - Tank
- The Manufacturer table contains the following values:
  - ACME
  - BURNS
  - SMITH

If the manufacturers ACME and BURNS both produce motors, in the Equipment Type System Code Table, you could add to the Motor System Code a reference to the *ACME* and *BURNS* System Codes. The references would indicate that the two manufacturers are associated with that equipment type.

Now assume that you have a family called *Motor* that will be used to store information about the motors in your company. Assume also that the Motor family contains the Manufacturer (ASSET_MANUF_CHR) field, which is intended to identify the name of the company that manufactured a given motor. In this case, you could use a Valid Values rule to create a list for this field that contains ALL values from the Manufacturer (MFGR) System Code Table. But since only *some* manufacturers produce motors, it would be better to filter the list to contain only the valid manufacturers: ACME and BURNS.

Assuming that the System Code references described above are already in place, you could use the Rules Wizard to implement this functionality by [generating a Valid Values rule](#) for the Manufacturer (ASSET_MANUF_CHR) field that is built from the Manufacturer System Code Table and contains a Static Filter to include only the values that are referenced by the Motor System Code in the Equipment Type (EQPT) System Code Table. When you are finished, the **Rule Details** screen should look like the following image.

The code that is generated for the GetPickList function will look like this:

```
-----------------------------------------------------------------------

Public Overrides Function GetPickList() As DynamicPickList
        Dim pickList As DynamicPickList
        'Use MyBase.CreatePickList(True)  if you want Restricted   PickList
        'Use MyBase.CreatePickList(False) if you want Unrestricted PickList
        pickList = MyBase.CreatePickList(True)
        PopulatePickList(pickList)
        Return pickList
End Function
Public Overrides Sub PopulatePickList(ByVal pickList As DynamicPickList)
        'Add an empty value to the top of the list
        'Delete the line [pickList.Add(DBNull.Value)] if you don't want the empty value at
        top of the list
        pickList.Add(DBNull.Value)
        pickList.AddActiveSystemCodes("MFGR")
End Sub


-----------------------------------------------------------------------
```

Filtering a System Code List Based on the Value in Another Field

Using System Codes, you have the ability to [create references](#) so that one System Code is referenced by another. After you have set up System Codes and references, you can use them to develop Valid Values rules.

For instance, consider an example where you have two System Code Tables: Manufacturer (MFGR) and Equipment Type (EQPT).

- The Manufacturer table contains the following values:
  - ACME
  - BURNS
  - SMITH
- The Equipment Type table contains the following values:
  - Compressor
  - Heat Exchanger
  - Motor
  - Pump
  - Tank

If the manufacturer *ACME* produces only motors and pumps, within the Manufacturer System Code Table, you could add a reference to the ACME System Code that references the Pump and Motor System Codes in the Equipment Type System Code Table. This would indicate that only the equipment types *Pump* and *Motor* are associated with the manufacturer *ACME*.

Now assume that you have a family called *Motor* that will be used to store information about the motors in your company. Assume also that the Motor family contains two fields: Equipment Type (ASSET_EQUIP_TYPE_CHR) and Manufacturer (ASSET_MANUF_CHR). In this case, you could use Valid Values rules to create lists for these fields that contain values from the Equipment Type (EQPT) and Manufacturer (MFGR) System Code Tables. But since only *some* equipment types are associated with each manufacturer, it would be more useful to create a rule where:

- The Manufacturer field contains a list of values from the Manufacturer System Code Table.
- The Equipment Type field contains a list of values from the Equipment Type System Code Table that are referenced by (i.e., are valid for) the selected manufacturer.

With these rules in place, if a user selects *ACME* in the **Manufacturer** list, the **Equipment Type** List will contain only *Pump* and *Motor*.

> ⚠ **Note:** When you are developing rules like this, keep in mind that the *direction* of the System Code references is important and will affect the behavior of the rules. If you created references *from* the Motor and Pump System Codes *to* the ACME System Code, this functionality would not be possible.

Assuming that the System Code references are already in place, you could use the Rules Wizard to implement this functionality by creating the following rules:

1. For the Manufacturer (ASSET_MANUF_CHR) field, generate a Valid Values rule that is built from the Manufacturer System Code Table and contains no filter. When you are finished, the **Rule Details** screen should look like the following image.

The code that is generated for the GetPickList function will look like this:

```
-------------------------------------------------------------------------

Public Overrides Function GetPickList() As DynamicPickList
        Dim pickList As DynamicPickList
        'Use MyBase.CreatePickList(True)  if you want Restricted   PickList
        'Use MyBase.CreatePickList(False) if you want Unrestricted PickList
        pickList = MyBase.CreatePickList(True)
        PopulatePickList(pickList)
        Return pickList
End Function
Public Overrides Sub PopulatePickList(ByVal pickList As DynamicPickList)
        'Add an empty value to the top of the list
        'Delete the line [pickList.Add(DBNull.Value)] if you don't want the empty
        value at top of the list
        pickList.Add(DBNull.Value)
        pickList.AddActiveSystemCodes("MFGR")
```

End Sub

----

2. For the Equipment Type (ASSET_EQUIP_TYPE_CHR) field, generate a Valid Values rule that is based upon the Equipment Type System Code Table and contains a Dynamic Filter that limits the list to values referenced from the Manufacturer (MFGR) System Code Table, based upon the value that is selected in the Manufacturer (ASSET_MANUF_CHR) field. When you are finished, the **Rule Details** screen should look like the following image.
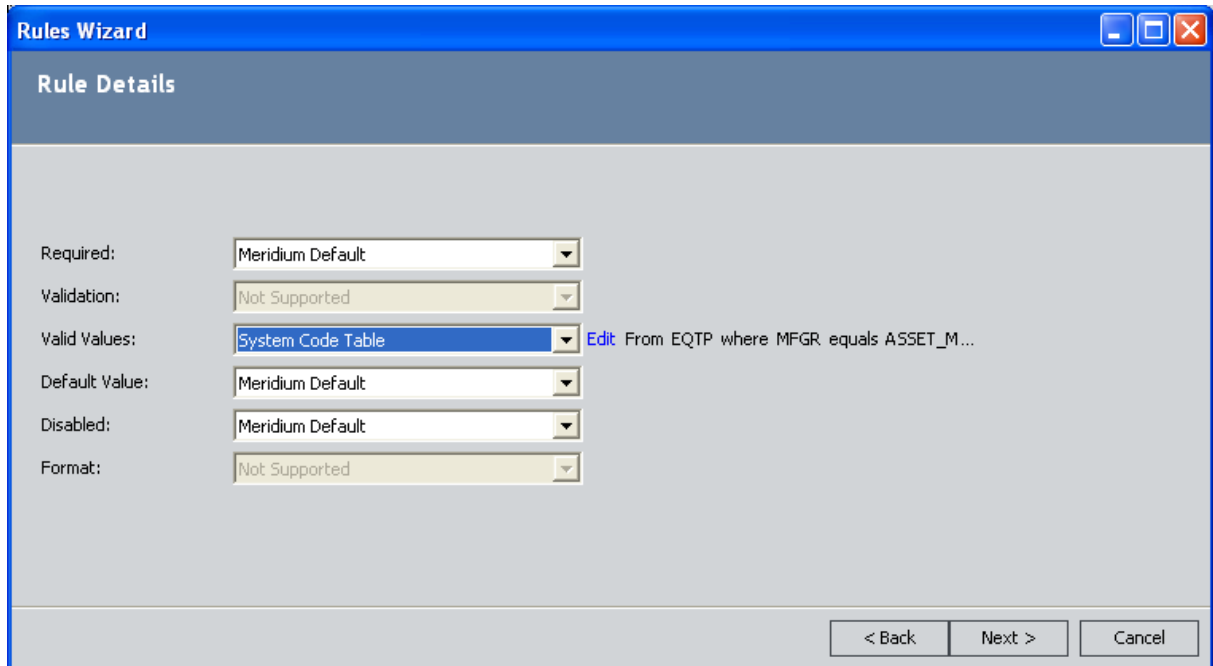
The rule that is generated for the GetPickList function will look like this:

```
Public Overrides Function GetPickList() As DynamicPickList
        Dim pickList As DynamicPickList
        'Use MyBase.CreatePickList(True)  if you want Restricted   PickList
        'Use MyBase.CreatePickList(False) if you want Unrestricted PickList
        pickList = MyBase.CreatePickList(True)
        PopulatePickList(pickList)
        pickList.RepopulateWhenChanges(CurrentDataRecord.Fields.Item("ASSET_
        MANUF_CHR"))
        Return pickList
End Function
Public Overrides Sub PopulatePickList(ByVal pickList As DynamicPickList)
        If Not Convert.IsDBNull(CurrentDataRecord.Fields.Item("ASSET_MANUF_
        CHR").Value) Then
                'Add an empty value to the top of the list
```

```
            'Delete the line [pickList.Add(DBNull.Value)] if you don't want the
            empty value at top of the list
            pickList.Add(DBNull.Value)
            pickList.AddActiveSystemCodes("EQTP", "MFGR", Cur-
            rentDataRecord.Fields.Item("ASSET_MANUF_CHR").Value)
        End If
    End Sub
```

Creating a Valid Values List from a Query

For some fields, it may be useful to construct a Valid Values list from query results. For instance, consider an example where the Work Order family contains the Equipment Technical ID field, which is meant to be populated with the technical numbers of the equipment for which the work is to be performed. In this case, you could construct a Valid Values list that contains the values in the Equipment Technical Number field in all the Equipment records in the database so that users can select the appropriate value in the Work Order record.

To implement this functionality, you would need to complete the following steps:

1. In the Meridium APM Framework application, create a Select query that includes the Equipment family and returns one column of values: Equipment Technical Number. For our example, assume that we have created the *Technical ID* query, which is stored in the Catalog folder \\Public\Meridium\Queries.

   > ⓘ **Note:** Users are not required to have permission to access the Catalog folder in which the query is stored in order to execute a rule that uses the query.

2. For the Equipment Technical ID field in the Work Order family, create a Valid Values rule that consists of the following code:

```
-------------------------------------------------------------------------

Public Overrides Function GetPickList() As DynamicPickList
        Dim pickList As DynamicPickList
        pickList = MyBase.CreatePickList(True)
        PopulatePickList(pickList)
        Return pickList
End Function

Public Overrides Sub PopulatePickList(ByVal pickList As DynamicPickList)

        Dim oCmd As Command
        Dim oRs As Meridium.Core.DataManager.Rowset
        Dim oRow As RowsetRow

        oCmd = New Command(CurrentEntity.ApplicationUser)
        oCmd.CommandText = "(!'Public\Meridium\Queries\Technical ID')"
        oRs = oCmd.Execute(RowsetMode.RawData)

        For Each oRow In oRs.Rows
                pickList.Add(oRow(0))
        Next oRow

End Sub


-------------------------------------------------------------------------
```

Note that the text shown in blue in this code identifies the name and location of the query. You can modify this text as needed to identify a different query. In this example, the specified query does not include prompts. To see an example of using a query that contains prompts, click here.

> ⓘ **Hint:** Our example uses a query that contains a *single* column of results. But the Valid Values rule actually specifies to build the list from the *first* column (the only column in our case) with the line **pickList.Add(oRow(0))**, where 0 (zero) identifies the first column. If your query contains more than one column of results, you can change this number to identify a different column (1 would identify the second column, a 2 would identify the third column, and so on).

Creating a Valid Values List from a Query that Contains Prompts

For some fields, it may be useful to construct a Valid Values list from a query that contains a prompt so that only a subset of the full results set is included in the list. [Consider our example of building a Valid Values list from a query that does not contain a prompt](#). Now, instead of including ALL Equipment Technical Numbers in the list, assume that you want to include only the Equipment Technical Numbers of Equipment records where the value in the Equipment Status field is *DLFL*.

To implement this functionality, you would need to complete the following steps:

1. In the Meridium APM Framework application, create a query that contains the Equipment Technical Number field and the Equipment Status field. On the Equipment Status column, create a prompt. You can create the prompt without any valid values. For our example, we will name the query *Technical Number by Status* and save it to the following folder in the Catalog: \\Public\Meridium\Queries.

2. For the Equipment Technical ID field in the Work Order family, create a Valid Values rule that consists of the following code:

```
------------------------------------------------------------------------

Public Overrides Function GetPickList() As DynamicPickList
        Dim pickList As DynamicPickList
        pickList = MyBase.CreatePickList(True)
        PopulatePickList(pickList)
        Return pickList
End Function

Public Overrides Sub PopulatePickList(ByVal pickList As DynamicPickList)

        Dim oCmd As Command
        Dim oRs As Meridium.Core.DataManager.Rowset
        Dim oRow As RowsetRow

        oCmd = New Command(CurrentEntity.ApplicationUser)
        oCmd.CommandText = "(!'Technical Number by Status')"
        oCmd.Parameters.Add(New CommandParameter(DataType.Character,
        "DLFL"))
        oRs = oCmd.Execute(RowsetMode.RawData)

        For Each oRow In oRs.Rows
                pickList.Add(oRow(0))
        Next oRow

End Sub

------------------------------------------------------------------------
```

Note that the text in blue above identifies the path and location of the query that you want to execute. The text in red identifies the value that will be passed in to the prompt. The line of code that contains the red text is the only difference between this rule and the one you would construct for a query that does not contain prompts. If the query contained more than one prompt, you would insert this line of code multiple times, once for each prompt, each time specifying the value to pass to the prompt.

Displaying the Current User as the Default Value

For some fields, you may want the default value to be set to the name of the user who is logged in at the time the record is created. For example, assume that the Work Order family contains the field *Created By*, which is intended to store the name of the user who created the record. In this case, you might want to create a Default Value rule on the Created By field to set it by default to the name of the user who is logged in to Meridium APM at the time the record is created. To do so, you could use the following rule code, which will populate the Created By field with the Meridium APM User ID of the current user.

Public Overrides Function GetDefaultInitialValue() As Object

  Return CurrentEntity.ApplicationUse.UserID

End Function

Displaying the Name of the Parent Family as the Default Value

Depending upon how your entity family hierarchy has been configured, it may be appropriate to populate fields in subfamilies with the name of their parent family. For example, consider a hierarchy that looks like this:

Equipment

      Pump

            Centrifugal Pump

Now, assume that the Centrifugal Pump family contains the field *Asset Type*, which is intended to categorize the records in that family according to the parent family. In this case, you could create the following rule on the Asset Type field to populate it with the value *Pump*.

```
Public Overrides Function GetDefaultInitialValue() As Object

    Dim fam As EntityFamily

    fam = CType(CurrentEntity.Family, EntityFamily)
    Return fam.Parent.Caption()

End Function
```

> 🛈 **Note:** The preceding rule will populate a field with the family caption of the family directly above that family (i.e., the *direct* parent).

Disabling a Field Based on the Value in Another Field

In some cases, it may be appropriate to disable a field conditionally, based upon the value in another field. For example, assume that you have a Work Order family that contains the *Completed By* and *Completed Date* fields. The Completed Date field is meant to indicate the date on which the work was completed, and the Completed By field is meant to indicate who completed it. The Completed Date field serves as a trigger to indicate that the work *has been completed*, so the Completed By field does not need to be enabled until the Completed Date field contains a value.

To enforce this logic, you could create the following rule on the Complete By field so that it is disabled when the Completed Date field is empty.

```
-----------------------------------------------------------------------

Public Overrides Function IsDisabled() As Boolean
        If Convert.IsDBNull(CurrentEntity.Fields("EWORK_COMPL_DATE_DT").Value) Or
        Object.Equals(CurrentEntity.Fields("EWORK_COMPL_DATE_DT").Value, "") Then
                Return True
        Else
                Return False
        End If
End Function


-----------------------------------------------------------------------
```

Formatting a Character Field as All Uppercase

Meridium APM does not allow you to generate formatting rules for character fields through the Rules Wizard. But there may be cases where you want to format character fields. For example, you may want to format character fields so that the value typed in the field appears in all capital letters. You could use the following Format rule to do so:

```
Public Overrides Function FormatValue(ByVal value As Object) As String
        If Not Convert.IsDBNull(CurrentEntity.Fields("AEMOT_EFFCL_CHR").Value) Then
                Return UCase(CStr(value))
        Else
                Return ""
        End If
End Function
```

In this example, AEMOT_EFFCL_CHR is the Field ID of the field that you want to format. This is the field on which the rule is defined.

> ⓘ **Note:** Like all Format rules, the rule shown above affects only the *displayed* value in places where Format rules are supported. The value will be *stored* in the database exactly as it is typed.

Formatting a Field to Display a Color Based on a Certain Value

Meridium APM does not allow you to generate formatting rules for character fields through the Rules Wizard. But there may be cases where you want to format character fields. For example, you may want to format a character field so that the value in the field causes the datasheet cell to be colored a certain color. You could use the following Format rule to do so:

```
Public Overrides Function FormatValue(ByVal value As Object) As String
        Dim Severity As String = Convert.ToString(value).ToUpper()
        If Severity.Contains("HIGH") Then
                CurrentDataField.BackColor = System.Drawing.Color.Red
        ElseIf Severity.Contains("LOW") Then
                CurrentDataField.BackColor = System.Drawing.Color.Green
        ElseIf Severity.Contains("MEDIUM") Then
                CurrentDataField.BackColor = System.Drawing.Color.Yellow
        End If
        Return value.ToString
End Function
```

In this example, the rule is defined on the Severity field. The rule dictates that:

- If the value in the Severity field is *HIGH*, the datasheet cell should be colored red.
- If the value in the Severity field is *LOW*, the datasheet cell should be colored green.
- If the value in the Severity field is *MEDIUM*, the datasheet cell should be colored yellow.

For example, consider the following record where the value in the Severity field is *HIGH*and the datasheet cell is colored red.

Note also that if a field is formatted in this way and you include it in a query that is running in *formatted* mode, the query results will also show the formatted color, as shown in the following image.

Assigned To	Due Date	Severity
John Smith	10/31/2011 12:00:00 AM	HIGH
Bill Johnson	10/19/2011 12:00:00 AM	LOW
Ellen James	10/28/2011 12:00:00 AM	MEDIUM

Calculating the Sum of Multiple Fields

For some fields, rather than requiring users to specify a value, you may want to display a value that is calculated from values entered in other fields. Consider an example where the Work Order family contains three cost fields:

- **Labor Cost (EWORK_LABCST_NBR):** The cost of the labor required to do the work.
- **Material Cost (EWORK_TOTMAT_NBR):** The cost of the parts or materials required to do the work.
- **Total Cost:** The total cost of the work, including the labor and materials cost.

In this case, you may want to require users to enter values in the Cost and Material Cost fields and then calculate the value for the Total Cost field by adding together the values in the Labor Cost and Material Cost fields. To implement this functionality, you could create the following Formula rule for the Total Cost field:

```
-----------------------------------------------------------------------

Public Overrides Function GetCalculatedValue() As Object
        Dim dTotal As Double
        dTotal = 0
        If Not Convert.IsDBNull(CurrentEntity.Fields("EWORK_LABCST_NBR").Value) Then
                dTotal = dTotal + CDbl(CurrentEntity.Fields("EWORK_LABCST_NBR").Value)
        End If
        If Not Convert.IsDBNull(CurrentEntity.Fields("EWORK_TOTMAT_NBR").Value) Then
                dTotal = dTotal + CDbl(CurrentEntity.Fields("EWORK_TOTMAT_NBR").Value)
        End If
        Return dTotal
End Function


-----------------------------------------------------------------------
```

> 🛈 **Note:** For this example to work properly, the Total Cost field must be set up as a [formula field](#).

Calculating the Difference Between Two Dates

Consider an example where the Work Order family contains three fields:

- **Out of Service Date (EWORK_OUTSERV_DT):** Specifies the date on which a piece of equipment was removed from service for maintenance.

- **Return To Service Date (EWORK_RETSERV_DT):** Specifies the date on which a piece of equipment returned to service after maintenance was completed.

- **Total Time to Repair (EWORK_TMETORPR_NBR):** The amount of time that it took to perform the maintenance.

Now, assume that you want users to specify a value in the Out of Service Date field and the Return to Service Date field and then calculate the value in the Total Time to Repair field as the difference between the two dates. You could implement this functionality by defining the following Formula rule on the Total Time to Repair field:

```
-----------------------------------------------------------------------

Public Overrides Function GetCalculatedValue() As Object

        Dim Date1 As Date
        Dim Date2 As Date

        Date1 = CDate(CurrentEntity.Fields("EWORK_OUTSERV_DT").Value)
        Date2 = CDate(CurrentEntity.Fields("EWORK_RETSERV_DT").Value)

        Return DateDiff(DateInterval.Day, Date1, Date2)

End Function


-----------------------------------------------------------------------
```

> ⓘ **Note:** For this example to work properly, the Total Time to Repair field must be set up as a [formula field](#).

Overview of the Rules Library

The Meridium APM Rules Library serves as a repository of rule projects that can be referenced from family rule projects. By storing rule code in the Rules Library and referencing it from family rule projects, you can reuse rule code repeatedly. This results in much more efficient and organized rule authoring than using family-level and field-level customization alone.

The Rules Library contains two types of projects:

- **Meridium APM Rule Projects:** Rule projects that are distributed as part of the Meridium APM baseline and are referenced within the Meridium APM baseline families. These projects are stored in the **Meridium APM** folder in the Rules Library.
- **Client Rule Projects:** Rule projects that are created by customers to support their own unique implementations of Meridium APM. These projects are stored in the **Client** folder in the Rules Library.

The topics in this section of the documentation are limited primarily to understanding and navigating the Rules Library interface. Information about rule code, including how to reference Rules Library projects from family projects is discussed elsewhere in this documentation.

> 📄 **Note:** To work with rule code in the Rules Library, you must have an understanding of VB.Net. Throughout this documentation, we limit our discussion to Meridium APM-specific aspects of the Rules Library, including information on using the Rules Library interface and guidelines that you must be aware of while you are working with rule projects. More information about VB.Net is available via the **Help** menu in VSTA.

Accessing the Rules Library

To access the Rules Library:

- In the Configuration Manager, on the main menu, click **Tools**, and then click **Manage Rules Library**.

  The **Rules Library** window appears.
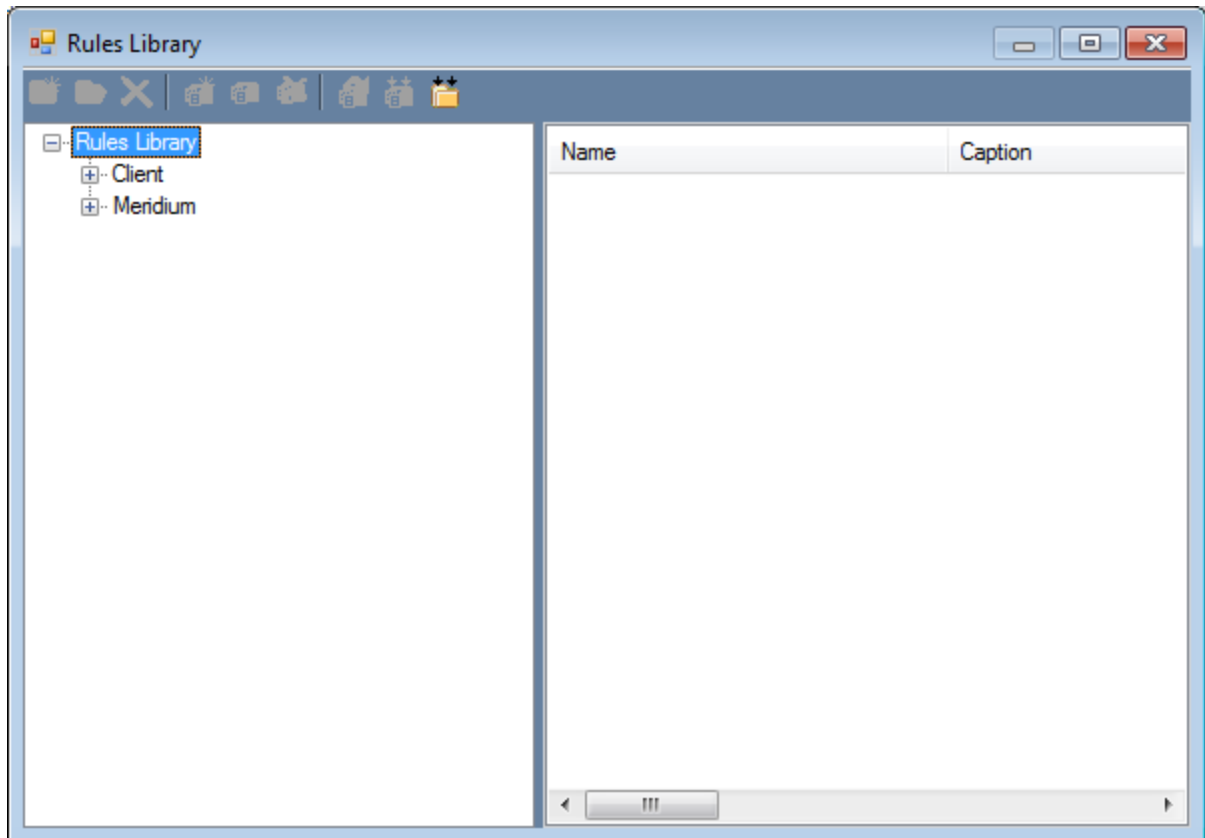
About the Rules Library Folder Structure

The **Rules Library** window in the Configuration Manager is divided into two main parts:

- The left pane, which displays the Rules Library folder structure.
- The right pane, which displays the rule projects that exist within each folder. When you select a folder in the left pane, its rule projects appear in the right pane.

The Rules Library folder structure contains two main folders within the **Rules Library** root folder:

- **Meridium APM:** Contains the rule projects that are delivered as Meridium APM product. The **Meridium APM** folder contains subfolders that organize the Meridium APM rule projects according to the Meridium APM module in which they are used. The subfolders include:

  - **Module-specific subfolders:** Folders that store rule projects, organized according to the module in which each project is used (e.g., Calibration, Inspection, Metrics, etc.). Note that these folders contain baseline rules, but they are not associated with a specific baseline family in the same way as the rule projects in the **Root Entity Families** folder. In other words, the **Calibration** rule project is not necessarily associated specifically with the Calibration family. It simply stores rule code that is used within Calibration Management.

  - **Root Entity Families:** Folders that store rule projects associated with specific, baseline entity families. The name of the rule project corresponds to the name of the family for which it stores rule code.

  - **Relationship Families:** Folders that store rule projects associated with specific, baseline relationship families. The name of the rule project corresponds to the name of the family for which it stores rule code.

  You can view the rule projects that are shipped with Meridium APM and reference them in any family you like, and you can copy any of the rule code and use it as the basis for creating your own rule projects. You cannot, however, modify or delete these projects, and you cannot add new projects to the Meridium folder. Projects in the Meridium folder will be updated automatically as needed for each Meridium APM upgrade.

- **Client:** Contains the rule projects that you have set up. The folder structure within the **Client** folder is entirely up to you. You can create as many subfolders as you like, and each subfolder can contain as many projects as you need.

> ⓘ **Note:** On SQL Server data sources, the **Client** folder appears above the **Meridium APM** folder in the folder structure.

This same folder and project structure is visible in the Meridium APM Framework application in the following Catalog folder: \\Public\Meridium\Rules Library.

> ⓘ **Note:** Only administrative users should be allowed to view this folder in the Meridium APM Framework application. Be sure to set up the Catalog folder permissions to restrict access to the **Rules Library** folder, as appropriate.

Options Available on the Rules Library Toolbar

A toolbar appears above the main display area on the **Rules Library** window to give you access to various functions. Each toolbar button is described in the following table.

Button	Name	Functions
	Create Folder	Lets you add a new folder to the Rules Library folder structure.
	Folder Properties	Displays the **Folder Properties** dialog box, where you can view the properties of the selected folder. Note that you can view properties only for subfolders in the **Client** folder. You must have the necessary privileges to view this information.
	Delete Folder	Deletes the selected folder and all the projects within the folder. You can also delete a folder and its contents by right-clicking the folder and choosing **Delete** on the shortcut menu. You cannot delete any folders in the **Meridium APM** folder.
	Create Project	Displays the **Project Properties** dialog box, where you can specify the name and description of a new rule project. You cannot create a new project in the **Meridium APM** folder.
	Project Properties	Displays the **Catalog Item Properties** dialog box, where you can view the properties of the selected rule project. Note that this button is enabled only when a rule project is selected in the right pane. You can also view the properties of a project by right-clicking it and choosing **Properties** on the shortcut menu.
	Delete Project	Deletes the selected rule project. Note that this button is enabled only when a rule project is selected in the right pane. You can also delete a project by right-clicking it and choosing **Delete** on the shortcut menu. You cannot delete any projects in the **Meridium APM** folder.
	Open Project	Opens the selected project in the **Meridium APM Rules Editor**. Note that this button is enabled only when a rule project is selected in the right pane. You can also open a project by right-clicking it and choosing **Open** on the shortcut menu.

Button	Name	Functions
	Compile Project	Displays the **Project Compilation** dialog box, where you can compile the selected project. Note that this button is enabled only when a rule project is selected in the right pane. You can also compile a project by right-clicking it and choosing **Compile** on the shortcut menu.
	Compile Folder	Displays the **Project Compilation** dialog box, where you can compile all the projects in the selected folder. You can also compile a folder by right-clicking it and choosing **Compile** on the shortcut menu.

Adding a Folder to the Rules Library Hierarchy

In the Rules Library, each rule project must be stored within a folder. You can add folders *only* under the **Client** folder.

To add a folder to the Rules Library hierarchy:

1. In the Configuration Manager, in the Rules Library folder hierarchy, select the **Client** folder, or select a subfolder of the **Client** folder.

2. On the toolbar, click **Create Folder** 📁 .

   The **Create Folder** dialog box appears.
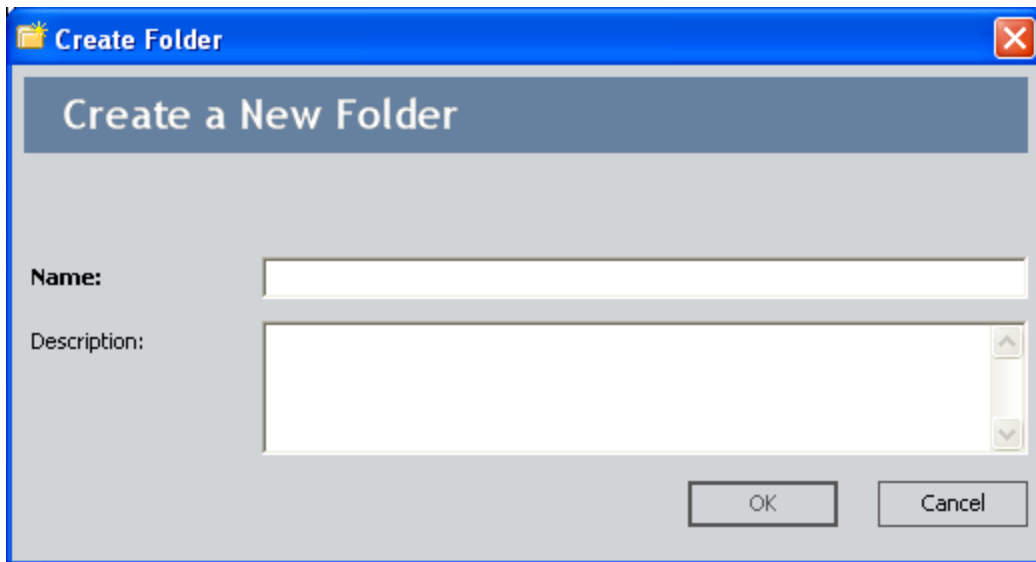
3. In the **Name** text box, type a name for the folder (required).

4. In the **Description** text box, type a description of the folder and the projects it will hold (optional).

5. Click **OK**.

   The folder is added to the Rules Library folder structure. After you have created the folder, you can add rule projects to it.

Deleting a Folder

To delete a folder and all the projects within that folder:

1. In the Configuration Manager, in the Rules Library folder hierarchy, select the folder that you want to delete.

2. On the toolbar, click **Delete Folder** ✖.

    A confirmation message appears, asking if you really want to delete the folder.

3. Click the **Yes** button.

    The folder and all the projects within it are deleted.

•

Creating a New Rule Project

To create a new rule project:

1. In the Configuration Manager, in the Rules Library folder hierarchy, select the folder to which you want to add the project.

   > ⓘ **Note:** You can add new rule projects *only* to the **Client** folder or one of its sub-folders.

2. On the toolbar, click **Create Project** 🏗.

   The **Project Properties** dialog box appears.

3. In the **Name** text box, type a name for the rule project (required).

   > ⓘ **Note:** The project name cannot begin with a number or contain spaces. If you specify a name that begins with a number, the name will be prepended automatically with an underscore. Additionally, any spaces in the project name will be replaced with underscores.

4. If desired, click the 🌐 button to manage the translations for that string.

5. In the **Description** text box, type a description of the project (optional).

6. Click **OK**.

   The system creates the new project and opens it automatically in the **Meridium APM Rules Editor** so that you can write the rule code.

7. Create the rule code as desired.

8. When you are finished, build the project and close the **Meridium APM Rules Editor**.

After you have created a new project you can compile it and then reference it from other projects.

•

Modifying an Existing Project

To modify an existing rule project:

1. In the Configuration Manager, in the Rules Library folder structure, select the folder that contains the project you want to modify.

2. In the pane on the right, select the project that you want to modify.

3. On the toolbar, click the **Open Project** button .

   The content of the rule project is displayed in the **Meridium APM Rules Editor**.

4. Modify the code as desired.

5. Build the project, and close the **Meridium APM Rules Editor**. If you have not saved the rule code prior to closing the **Meridium APM Rules Editor**, you will be prompted to do so.

Deleting a Rule Project

To delete a rule project:

1. In the Configuration Manager, in the Rules Library folder hierarchy, select the folder that contains the rule project want to delete.

2. In the pane on the right, select the rule project.

3. On the toolbar, click the **Delete Project** button ![delete icon] .

   A confirmation message appears, asking if you really want to delete the project.

4. Click the **Yes** button.

   The project is deleted.

•

About Making Calls to the Rules Library

Creating Rules Library projects is only the first step in implementing family and field customizations using the Rules Library. After the desired Rules Library project has been created, it must be called from the family or field class of the family or field by which it will be used. You must complete two main steps to call a Rules Library project from a family project:

1. First, you must add a reference to the Rules Library project from the family project. This will make the rules that are stored within the Rules Library project available for use within the family rule project.

2. Second, within the family rule project, you must make calls to the parts of the referenced Rules Library project that you want to use. How and where you make the calls will depend upon how the Rules Library project is organized and where you want to invoke certain functionality.

After the Rules Library project has been called within the family rule project, it can be extended via the family rule project in order to customize the rule for that specific family.

Adding a Reference to a Rules Library Project

Because Rules Library projects are external to family rule projects, you must create a reference to a Rules Library project before it can be used in a family rule project. You can also add references to Rules Library projects within *other* Rules Library projects.

> **Note:** You can create references to **Meridium APM** rule projects *and* to **Client** rule projects.

To create a reference to a rule project:

1. In the Configuration Manager, open the family rule project where you want to add a reference. You can open an existing rule project, or you can open family rule code (i.e., business rules).

   For example, the following figure shows what would be displayed in the **Meridium APM Rules Editor** if you accessed the rules for the MI_Thickness_Measurement family.
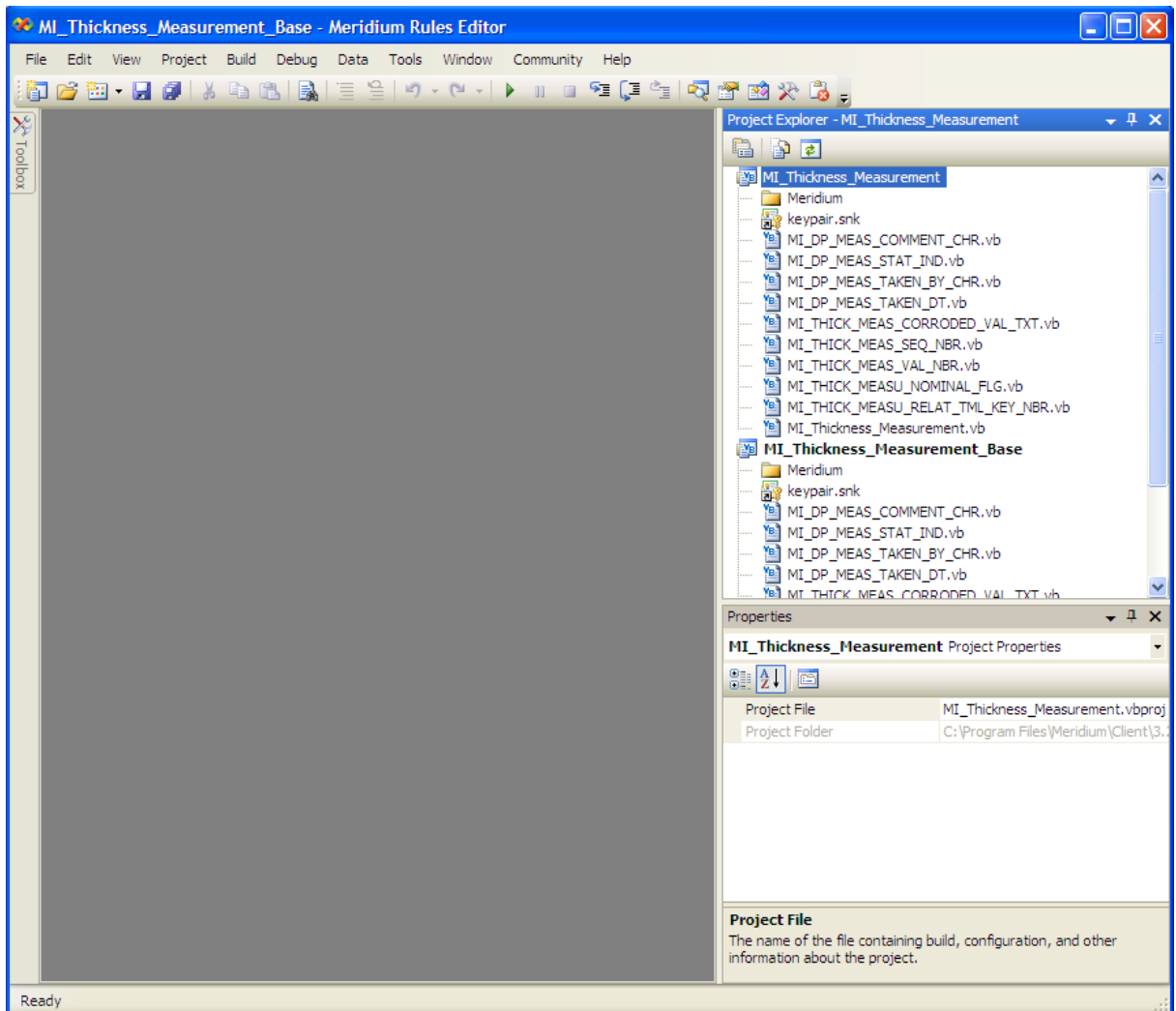
2. In the right pane, right-click **MI_Thickness_Measurement**, and then click **Add Reference**.

> ⓘ **Hint:** You can click the **Show All Files** icon 📄 to view references in the list of files that belong to the project.

The **Add Reference** window appears.

3. On the **.NET** tab, scroll down in the list until you find the rule project that you want to reference. Rule project names use the following syntax:

Library.[Folder].[Project Name]

Where **[Folder]** is the name of the folder in which the project resides (i.e., **Meridium APM** or **Client**) and [**Project Name**] is the name of the project itself.

> 🗒 **Note:** New projects must be compiled before they will appear in the list of available references.

4. Highlight the desired project, and click **OK**.

The **Add Reference** window closes and the selected project appears in the list of **References** in the **Project Explorer** pane.

5. If desired, repeat steps 2 through 4 to add additional project references.

6. Compile the project or family to which you added the reference(s).

How To Call Rules Library Projects

How you make calls to the Rules Library projects that are referenced from family projects will depend upon how the Rules Library project is organized, what specific functionality you want to invoke, and when you want to invoke it. There are numerous ways in which to organize your Rules Library projects. How you structure them is up to you. In this area of the documentation, we discuss some of the most common ways to organize Rules Library projects and then invoke the functionality within them.

> **Note:** We do not discuss every possible method of organization or invocation. In addition, you may choose to use a combination of the organization options that are available.

To invoke functionality defined within a Rules Library project, you might:

- Use an Inherits statement to inherit from a field class the functionality stored within a Rules Library class.
- Call a specific function that is defined within a Rules Library project.

Inheriting a Rules Library Class from a Field Class

One way to organize a Rules Library project is to create family and field classes within the project that will then be referenced by family and field classes within family rule projects. This is how the baseline Meridium APM rules are organized, and it serves as a valid and useful method for organizing your custom rules as well. One major benefit of this method of organization is that it allows you to reuse the same base class for multiple families but extend the basic rules for certain families, as needed.

For example, maybe your database contains various custom families with valid values lists. One way to organize your Rules Library projects would be to create a ValidValuesLists project. Within the ValidValuesLists project, you could create a class for each type of valid values list that you plan to reuse across multiple families.

For example, consider the following class (called *Status*), which is defined in the ValidValuesLists project. This logic populates a static list with the values *Evaluating*, *In Progress*, and *Complete*.

```
------------------------------------------------------------------

Option Strict On
Option Explicit On

Imports Meridium.Core.DataManager
Imports Meridium.Core.DataManager.Customization
Imports Meridium.Core.Internals
Imports Meridium.Core.Metadata.MetadataReader
Imports Meridium.Core.Security.ApplicationUser
Imports Meridium.Core.UOM
Imports System
Imports System.Xml
Public Class YesNoMaybe
        Inherits Meridium.Core.DataManager.Customization.EntityFieldCustomization
        Public Sub New(ByVal record As Meridium.Core.DataManager.DataRecord, ByVal
        field As Meridium.Core.DataManager.DataRecord
                MyBase.New(record, field)
        End Sub
        Public Overrides Function GetPickList() As DynamicPickList
                Dim pickList As DynamicPickList
                'Use MyBase.CreatePickList(True) if you want Restricted PickList
                'Use MyBase.CreatePickList(False) if you want Unrestricted PickList
                pickList = MyBase.CreatePickList(True)
                PopulatePickList(picklist)
                Return pickList
        End Function
        Public Overrides Sub PopulatePickList(ByVal pickList As DynamicPickList)
                pickList.AddRange("Evaluating|In Progress|Complete".Split("|"c))
        End Sub
End Class
```

To invoke this valid values rule within a given field, you would simply add a reference to the **ValidValuesList** project and then use an Inherits statement to call the **Status** class within the desired family field class. For example, the following code shows this rule being inherited by the **Purchase Order Status** field.

```
<MetadataField("PURCH_ORDER_STATU_CHR")> _
Public Class PURCH_ORDER_STATU_CHR
        Inherits Library.Client.ValidValuesLists.Status

        Public Sub New (ByVal record As Meridium.Core.DataManager.DataRecord, ByVal
        field As Meridium.Core.DataManager.DataRecord
             MyBase.New(record, field)
        EndSub

End Class
```

Like the baseline Meridium APM rules, Rules Library projects that are organized and called in this way can be extended at the field level. For example, the following code adds the values **On Hold** and **Rejected** to the list constructed by the Rules Library class.

```
<MetadataField("PURCH_ORDER_STATU_CHR")> _
Public Class PURCH_ORDER_STATU_CHR
        Inherits Library.Client.ValidValuesLists.Status

        Public Sub New (ByVal record As Meridium.Core.DataManager.DataRecord, ByVal
        field As Meridium.Core.DataManager.DataRecord
             MyBase.New(record, field)

        Public Overrides Sub PopulatePickList(ByVal pickList As DynamicPickList)
             MyBase.PopulatePickList(pickList)
             pickList.Add("On Hold")
             pickList.Add("Rejected")
```

Calling a Function or Sub Procedure Stored in a Rules Library Project

Another way to organize Rules Library projects is to define custom functions and sub-procedures within a single Rules Library class and then call those functions and sub-procedures from family classes. For example, consider the following simple Rules Library class called **CustomBehaviors**, which exists within a Rules Library project of the same name.

> ℹ **Note:** Although the following example is very simple, keep in mind that typically you will use the Rules Library to store complex code that you want to maintain in a single place and use across multiple families. For examples of more complex code, see the section "Sample Code" in this documentation.

```
------------------------------------------------------------------------

Option Strict On
Option Explicit On

Imports Meridium.Core.DataManager
Imports Meridium.Core.DataManager.Customization
Imports Meridium.Core.Internals
Imports Meridium.Core.Metadata.MetadataReader
Imports Meridium.Core.Security.ApplicationUser
Imports Meridium.Core.Uom
Imports System
Imports System.Xml

Public Class CustomBehaviors
    Public Sub New()
        MyBase.New()
    End Sub

    Public Shared Function DefaultValue() As Object
        Return CStr("Default Value")
    End Function

    Public Shared Sub Statuses(ByRef picklist As DynamicPickList)
        picklist.Add("Status1")
        picklist.Add("Status2")
        picklist.Add("Status3")
    End Sub
End Class

------------------------------------------------------------------------
```

Within this Rules Library class, there is one function (**DefaultValue**) and one sub-procedure (**Statuses**). After these items have been defined in the Rules Library, they can be called from family rule projects. The first step is always to add a reference to the Rules

Library project. The following excerpt shows how to call the **DefaultValue** function from a family field class.

```
------------------------------------------------------------------------

Option Strict On
Option Explicit On

Imports Meridium.Core.DataManager
Imports Meridium.Core.DataManager.Customization
Imports Meridium.Core.Internals
Imports Meridium.Core.Metadata.MetadataReader
Imports Meridium.Core.Security.ApplicationUser
Imports Meridium.Core.Uom
Imports Library.Client.CustomBehaviors.CustomBehaviors
Imports System
Imports System.Xml

<MetadataField("FAMIL_FIELD_1_CHR")> _
Public Class FAMIL_FIELD_1_CHR
    Inherits Meridium.Core.DataManager.Customization.EntityFieldCustomization

Public Sub New(ByVal record As Meridium.Core.DataManager.DataRecord, ByVal field As
Meridium.Core.DataManager.DataField)
        MyBase.New(record, field)
    End Sub

    Public Overrides Function GetDefaultInitialValue() As Object
        Return CStr(DefaultValue())
    End Function
End Class

------------------------------------------------------------------------
```

Note that:

- An *Imports* statement has been added at the beginning of the class to reference the Rules Library class in which the function is defined. This is *in addition to* the [reference you must add to the rule project itself](#).
- The text shown in bold is the actual call to the *DefaultValue* function. When called in this way, the value defined for the DefaultValue function (i.e., Default Value) will be used as the default value for this field.

The following portion of code shows how the Statuses sub-procedure in the same Rules Library project can be called from another field class.

```
------------------------------------------------------------------------

Option Strict On
Option Explicit On
```

```
Imports Meridium.Core.DataManager
Imports Meridium.Core.DataManager.Customization
Imports Meridium.Core.Internals
Imports Meridium.Core.Metadata.MetadataReader
Imports Meridium.Core.Security.ApplicationUser
Imports Meridium.Core.Uom
Imports Library.Client.CustomBehaviors.CustomBehaviors
Imports System
Imports System.Xml

<MetadataField("FAMIL_FIELD_2_CHR")> _
Public Class FAMIL_FIELD_2_CHR
   Inherits Meridium.Core.DataManager.Customization.EntityFieldCustomization

Public Sub New(ByVal record As Meridium.Core.DataManager.DataRecord, ByVal field As
Meridium.Core.DataManager.DataField)
     MyBase.New(record, field)
   End Sub

  Public Overrides Function GetPickList() As DynamicPickList
      Dim pickList As DynamicPickList
      pickList = MyBase.CreatePickList(True)
      PopulatePickList(pickList)
      Return pickList
   End Function
   Public Overrides Sub PopulatePickList(ByVal pickList As DynamicPickList)
      Statuses(pickList)
      Return
   End Sub
End Class
```

Like the previous example, note that:

- An *Imports* statement has been added at the beginning of the class to reference the Rules Library class in which the sub-procedure is defined. This is *in addition to* the reference you must add to the rule project itself.
- The text shown in bold is the actual call to the *Statuses* sub-procedure. When called in this way, the value defined for the Statuses sub-procedure will be used to populate the drop-down list for this field.

Introduction to Baseline Rule Storage

All baseline rules are stored in the Meridium APM Rules Library and inherited by the families by which they are used. This method of storage offers several advantages over storing baseline rules in the family projects themselves:

- It allows you to extend, or customize, the baseline functionality in the same way that you would extend any Rules Library functionality through family-level or field-level customization.

- If desired, it lets you easily disable the baseline functionality by changing one line of code: the inheritance statement that references the Rules Library project.

- It allows Meridium, Inc. to protect its baseline content from one version to the next. Because the baseline rules are stored in the Rules Library in a location where they cannot be modified or deleted by customers, Meridium, Inc. can easily deliver updates to that code during an upgrade.

The Structure of Baseline Rules Library Projects

In the Meridium APM Rules Library, within the Meridium folder, the Root Entity Families and Relationship Families folders store rule projects associated with the baseline Meridium APM families. Each rule project is referenced from a Meridium APM family, whose family ID corresponds to the name of the rule project.

The organization of the baseline family Rules Library projects resembles that of the family structure in the **Entity Family/Relationship Family** pane of the Configuration Manager main window but does not match it exactly. When you select the Root Entity Families folder in the Rules Library, a list of rule projects appears in the right pane of the **Rules Library** window. These projects correspond to the families directly *under* the Root-level family.

For example, the following image shows the MI Datapoint Measurement project, which is circled in red.
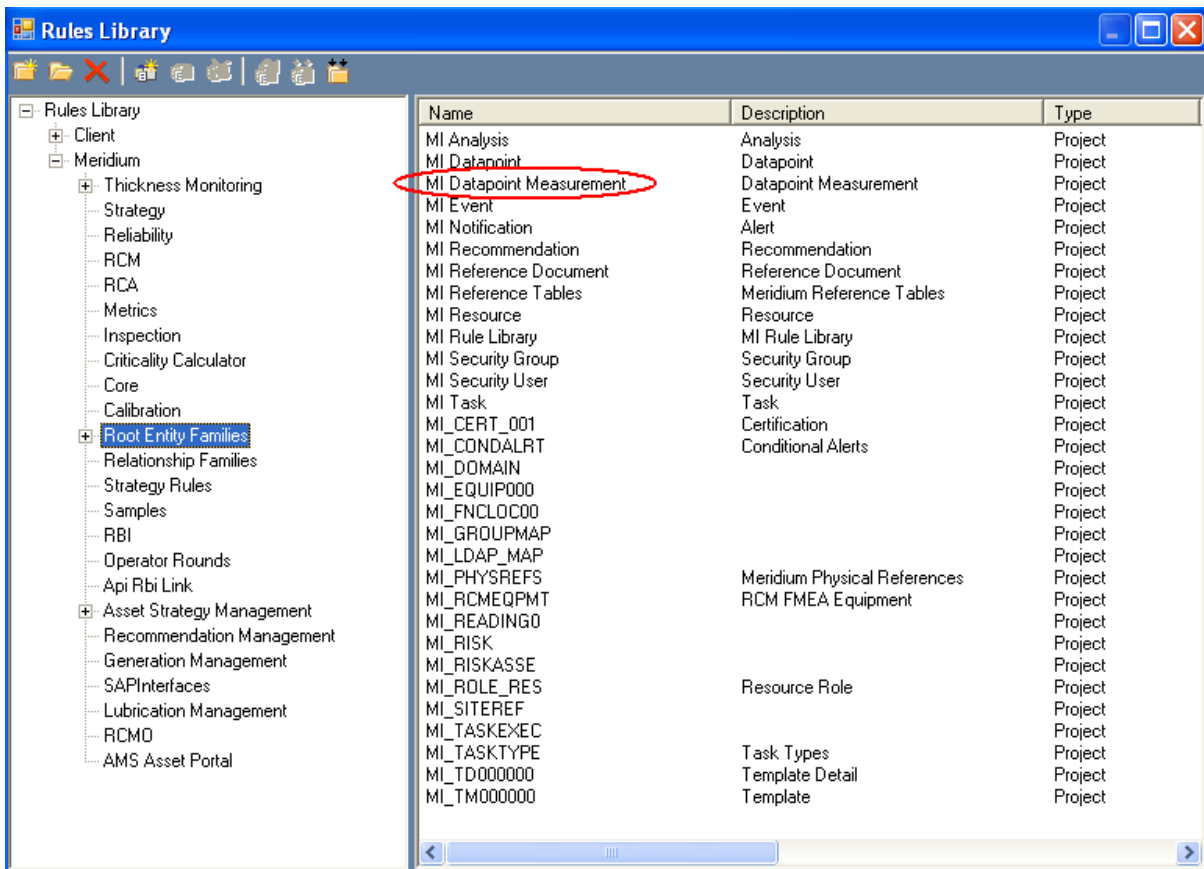
The MI Datapoint Measurement project stores the rule code for the Datapoint Measurement family, which is circled in red in the following image.
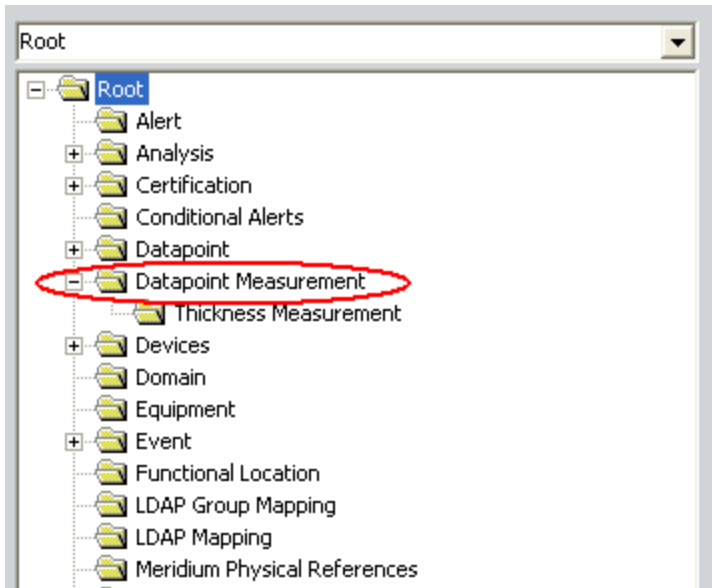
Just as the name of the *family* rule project matches the family ID of a given family, the name of the corresponding Rules Library project also corresponds to the family ID.

- The project name that appears in the **Rules Library** window is exactly the *family ID* (e.g., MI Datapoint Measurement).

- The *Description* that appears to the right of the project name in the **Rules Library** window matches the family *caption* (e.g., Datapoint Measurement).

- The actual name of rule project (i.e., when you open the project in VSTA) is the family ID with **_Base** appended to it, where any spaces have been replaced by under-scores (e.g., MI_Datapoint_Measurement_Base).

> ⓘ **Hint:** *Base* identifies the rule project as the baseline rule project. It also helps distinguish the *baseline* rule project from the *family* rule project when you open the family rules in the VSTA.

If you expand the Root Entity Families folder, you will see that it contains subfolders. These subfolders store projects for subfamilies of the root-level families. For example, the MI Datapoint Measurement folder contains one project: MI Thickness Measurement.
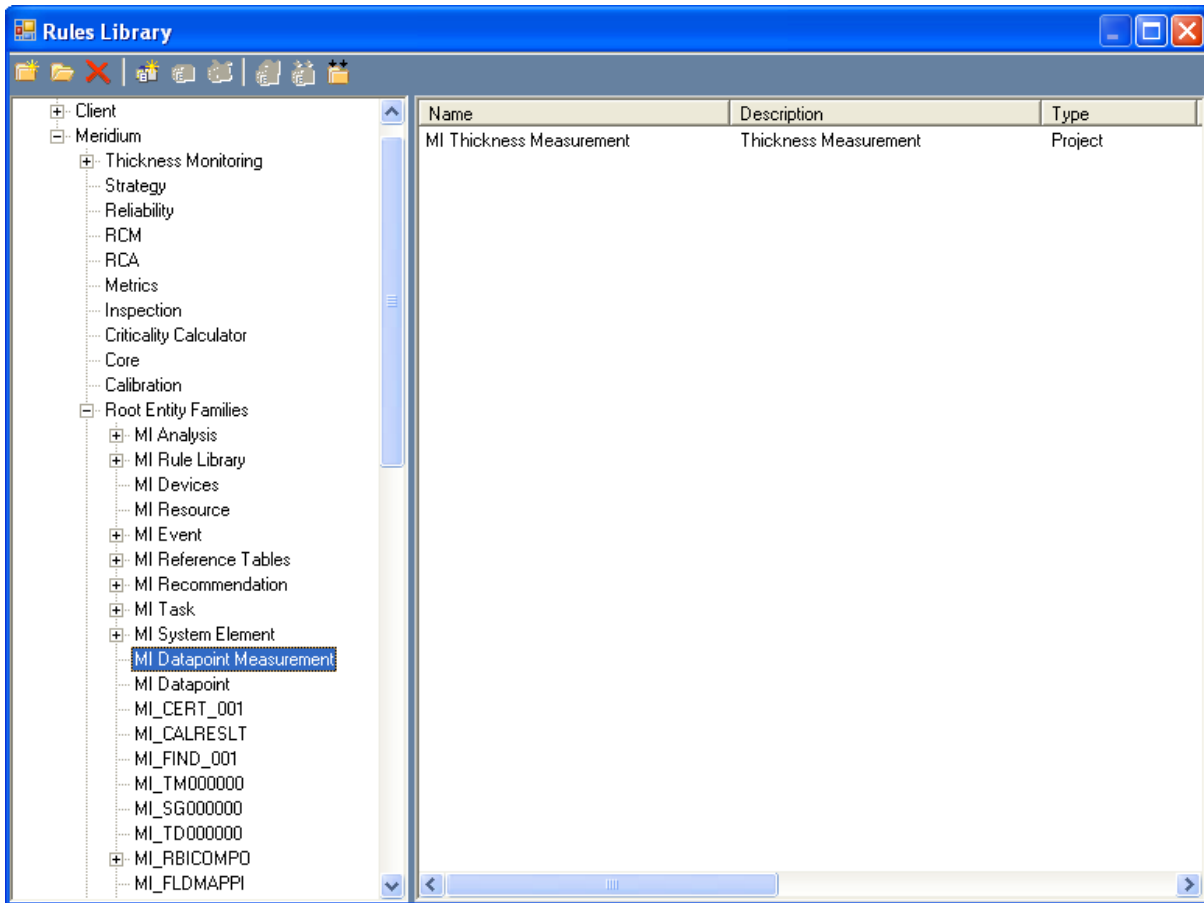
These projects correspond to the child family of the Datapoint Measurement family:
Thickness Measurement.
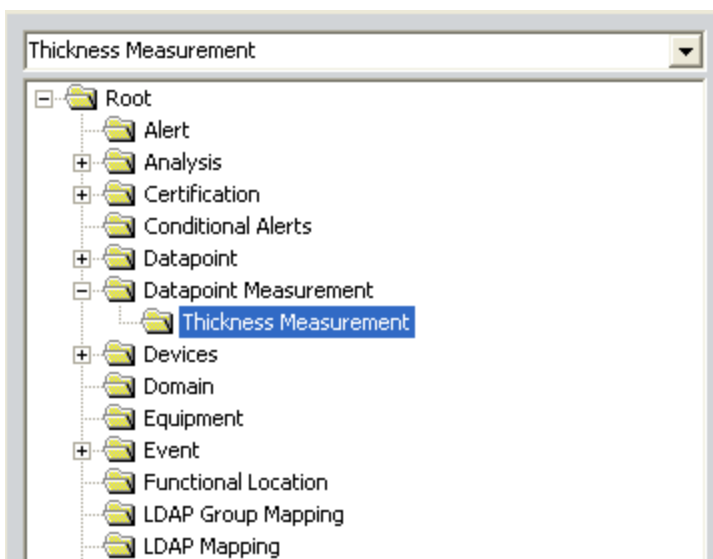
All subfolders within the Root Entity Families folder are organized in this way, where each project folder stores Rules Library projects for a given set of subfamilies. The project *folde*r name matches the family ID of the family, and the *project* names correspond to the family IDs of the subfamilies.

> ⓘ **Note:** The Relationship Families folder is organized in a similar way, except that all projects are defined at the Relationship Families level, as all relationship families are defined at the root level.

Note that when you open the Rules Library, you may not see a baseline Rules Library project for every baseline Meridium APM family. Initially, baseline Rules Library projects exist only for families that are shipped with *actual* baseline rule code. If baseline rule code is not shipped for a given family, the baseline Rules Library project will be created at the same time that the family rule project is created (i.e., when you access the family rules).

The Content of Baseline Rule Projects

Each baseline Rules Library project contains files that correspond to the files that exist in the associated baseline family rule project. For example, consider the family rule project for the baseline Datapoint Measurement family (MI Datapoint Measurement) shown in the following image.
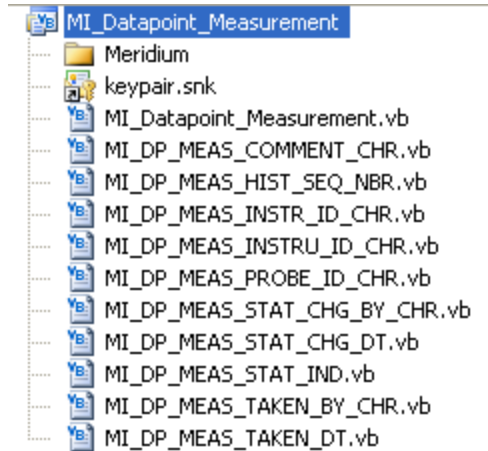
Note that 11 files exist in the family rule project: one for each of the 10 fields that are defined in the baseline Datapoint Measurement family and one for the family itself. Now, consider the corresponding baseline Rules Library project shown in the following image.
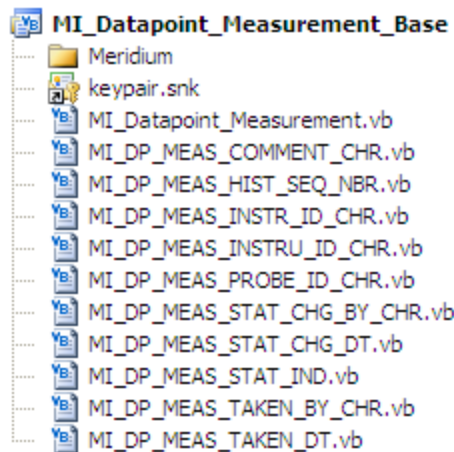
Note that it, too, contains 11 files, each of which corresponds to a file that is defined in the baseline *family* rule project. And like the family rule project, the files in the baseline Rules Library project store classes that contain rule code for the associated family or field. The rule code within each baseline Rules Library class corresponds directly to the rule code that was stored in the baseline *family* classes prior to the V3.2.0. In other words, the baseline Rules Library project is an exact copy of what was previously the baseline family project. Each class in the baseline Rules Library project is inherited by

[the corresponding class](#) in the family rule project to invoke the rule that is stored in the Rules Library.

> 📄 **Note:** The baseline Rules Library project always contains a baseline class for *each* family and field class, even if the baseline class itself does not contain any actual rule code. Likewise, each family and field class *always* inherits from the baseline Rules Library class, even if no rule code exists in that class. In cases where the baseline Rules Library class does not contain any rule code, it exists as a holding area where baseline Meridium APM rules may be added to that family in future releases.

How Baseline Rules Are Called Within Family Projects

Baseline Rules Library projects are called from their corresponding family rule projects using an *Inherits* statement. For example, the baseline Root Cause Analysis data model includes a family called *RCA Analysis*.
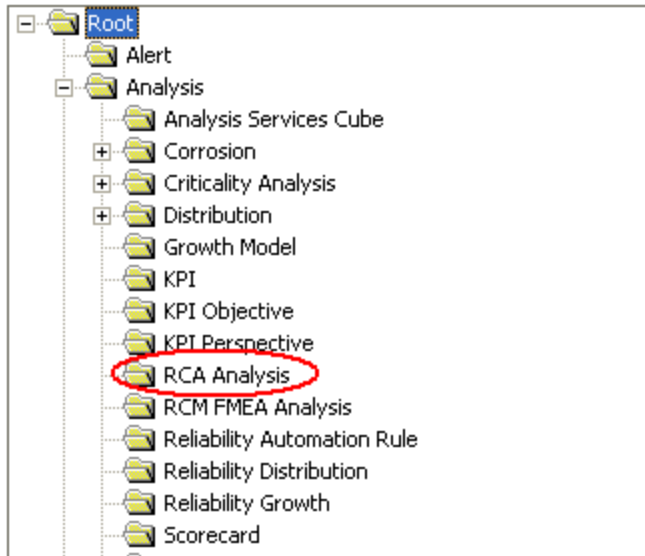
The baseline rules for the RCA Analysis family are stored in the *MI RCA Analysis* rule project.
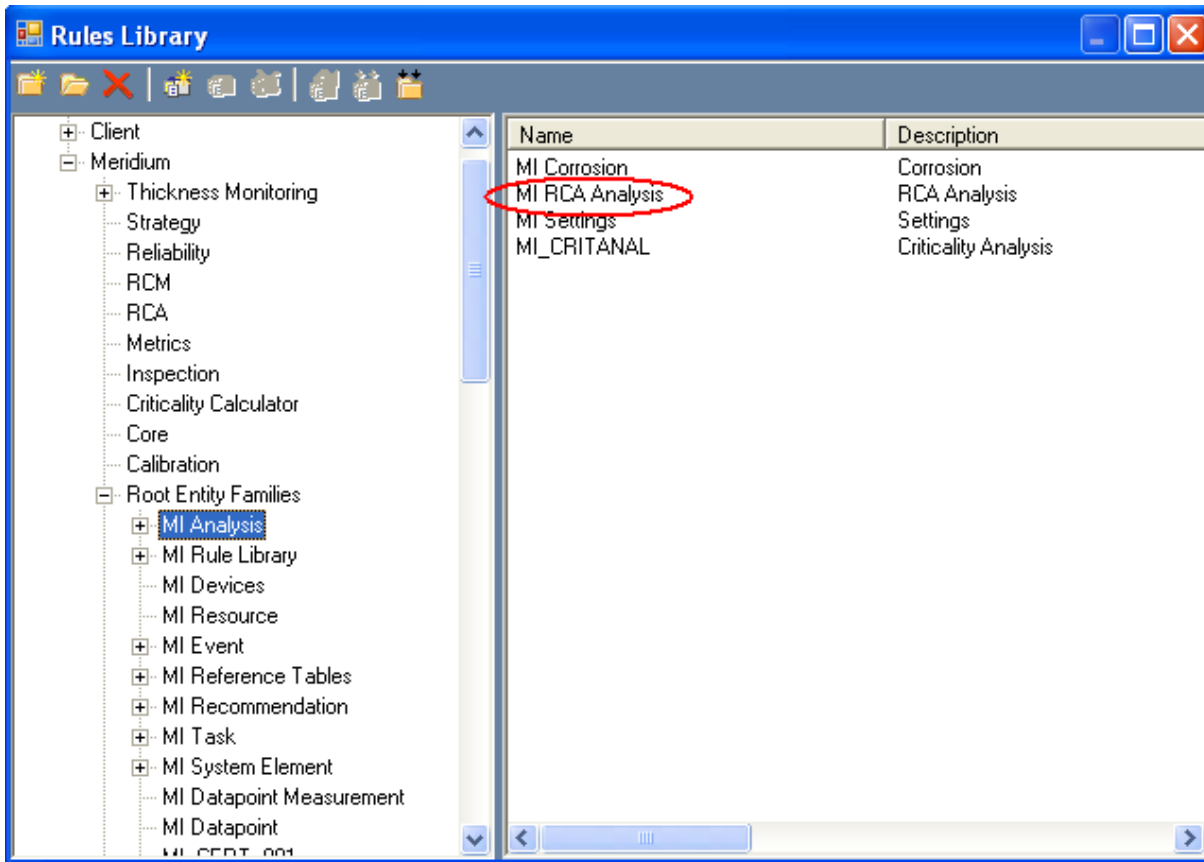
If you open the family rule project for the RCA Analysis family, you will see that the MI_RCA_Analysis rule project (i.e., the family rule project) contains a reference to the MI_RCA_Analysis_Base project (i.e., the baseline Rules Library project for the RCA Analysis family).

> **Note:** When you open the family rule project for the RCA Analysis family, note that the MI_RCA_Analysis_Base project also opens in the **Project Explorer** pane. This gives you easy access to the content of the baseline rule project so that you can see what specific rules will be invoked when calls are made from the family project to code in the baseline rules project.

If you open the MI_RCA_Analysis file in the MI_RCA_Analysis project, you will see the Inherits statement that is circled in the following image.

•

```vb
Option Strict On
Option Explicit On

Imports Meridium.Core.DataManager
Imports Meridium.Core.DataManager.Customization
Imports Meridium.Core.Internals
Imports Meridium.Core.Metadata.MetadataReader
Imports Meridium.Core.Security.ApplicationUser
Imports Meridium.Core.Uom
Imports System
Imports System.Xml


<MetadataFamily()> _
Public Class MI_RCA_Analysis
    Inherits Baseline.MI_RCA_Analysis.MI_RCA_Analysis


    Public Sub New(ByVal record As Meridium.Core.DataManager.DataRecord)
        MyBase.New(record)
    End Sub
End Class
```

This *Inherits* statement specifies that the MI_RCA_Analysis class in the *family* rule project will inherit the rule defined in the MI_RCA_ANALYSIS class in the MI_RCA_Analysis_Base rule project. Each class within the MI_RCA_ANALYSIS family rule project for which a corresponding class exists in the MI_RCA_ANALYSIS_Base rule project will have a similar Inherits statement, specifying that the field rule should be inherited from the class in the Rules Library project. You can open the MI_RCA_Analysis_Base project to see the actual rule code that will be applied to each field through the inheritance statement.

Every baseline family rule project contains a reference to the corresponding baseline Rules Library project. In addition, every class within the baseline family project inherits from the corresponding Rules Library class. Note that this does not mean that every field in the baseline Meridium APM data model has baseline rules defined for it. Not all baseline Rules Library classes contain actual rule code.

> **Note:** Any class that is inherited by another class is considered a *base class* of the class that inherits it. This means that each baseline Rules Library class functions as a *base class* of the corresponding family or field class.

Levels of Customization

The Meridium APM method of implementing baseline rules through the Rules Library offers you flexibility in how you customize family rules. In general, you can choose one of three levels of customization for each baseline family and field:

- **No customization:** You do not want to customize the baseline rule at all. Instead, you want to apply the default rules supplied by Meridium APM.
- **Some customization:** You want to keep the default Meridium APM rules but extend them or override part of them.
- **Full customization:** You do not want to keep any of the baseline Meridium APM rules. You want the family or field to be controlled entirely by custom code.

Within a family rule project, customization is done on a class-by-class basis. In other words, you might accept the baseline rules for some classes, extend the rules of others, and fully customize others. Your database will likely contain a combination of customization types.

Maintaining Baseline Functionality

In the baseline Meridium APM rules implementation, all baseline family and field classes inherit from a corresponding baseline Rules Library class. This causes the Rules Library rule to be invoked by the corresponding family or field. Therefore, to maintain the baseline Meridium APM functionality in the current version and in future versions, you would simply *keep the baseline inheritance* and not modify the code within the baseline family or field class.

Any changes delivered to the baseline rule in a future release would be inherited automatically by the family or field. No additional action would need to be taken to apply the default rule.

Extending the Baseline Functionality

If you want to use the baseline functionality as a starting point for how a family or field should behave, you can customize the family rule project to *extend* the baseline rule. To do this, you would keep the default Inherits statement, call the baseline Rules Library functions that you want to use, and then write code to extend those rules.

For instance, consider for the purposes of this example that the baseline Meridium APM data model contains a family called **Example Family**. Now, assume that the **Example Family** contains a field called **Example Field 1**, which might have a baseline Rules Library project that looks something like this:

```
Imports System
Imports Meridium.Core.DataManager
Imports Meridium.Core.DataManager.Customization
Imports Meridium.Core.Uom
Imports Meridium.Core.Internals
Imports Meridium.Core.Metadata

<MetadataField("EXAMP_FAMIL_EXAMP_FIELD_1_CHR")> _
Public Class EXAMP_FAMIL_EXAMP_FIELD_1_CHR
    Inherits EntityFieldCustomization

#Region "Automatically generated code, do not modify!"
    Public SubNew(ByVal record As DataRecord, ByVal field As DataField)
        MyBase.New(record, field)
    End Sub
#End Region

    Public Overrides Function GetPickList() As DynamicPickList
        Dim pickList As DynamicPickList
        'Use MyBase.CreatePickList(True)  if you want Restricted   PickList
        'Use MyBase.CreatePickList(False) if you want Unrestricted PickList
        pickList = MyBase.CreatePickList(True)
        PopulatePickList(pickList)
        Return pickList
    End Function
    Public Overrides Sub PopulatePickList(ByVal pickList As DynamicPickList)
        pickList.Add("A")
        pickList.Add("B")
    End Sub
End Class
```

This code specifies that in an Example Family record, the field **Example Field 1** will contain a drop-down list with the values **A** and **B**. In the baseline implementation, the corresponding field class will inherit from the Rules Library class so that when a user creates an **Example Family** record, the field **Example Field 1** will display a drop-down list containing values **A** and **B**.
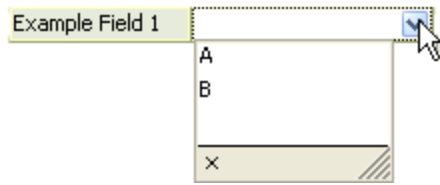
Now, suppose that you want to *maintain* the functionality provided by baseline implementation so that the list will contain values **A** and **B** but that you *also* want the list to contain values **C** and **D**. By customizing the field class, you can add values **C** and **D** to the list created by the baseline implementation. For example, consider the following field customization.

```
Option Strict On
Option Explicit On

Imports System
Imports Meridium.Core.DataManager
Imports Meridium.Core.DataManager.Customization
Imports Meridium.Core.Uom
Imports Meridium.Core.Internals
Imports Meridium.Core.Metadata

<MetadataField("EXAMP_FAMIL_EXAMP_FIELD_1_CHR")> _
Public Class EXAMP_FAMIL_EXAMP_FIELD_1_CHR
    Inherits Baseline.Example_Family.EXAMP_FAMIL_EXAMP_FIELD_1_CHR

#Region "Automatically generated code, do not modify!"
    Public SubNew(ByVal record As DataRecord, ByVal field As DataField)
        MyBase.New(record, field)
    End Sub
#End Region

    Public Overrides Sub PopulatePickList(ByVal pickList As DynamicPickList)
        MyBase.PopulatePickList(pickList)
        pickList.Add("C")
        pickList.Add("D")
    End Sub
End Class
```

In this example, the inheritance has not changed, so the field continues to inherit from the baseline Rules Library class. Then line **MyBase.PopulatePickList(pickList)** calls the **PopulatePickList** function from the baseline Rules Library project, which adds values **A** and **B** to the list. The subsequent lines **pickList.Add("C")** and **pickList.Add("D")** add values**C**and**D**to the list as well. As a result, when a user creates an Example Family record, the field**Example Field 1** will display a drop-down list containing the values **A**, **B**, **C**, and **D**.
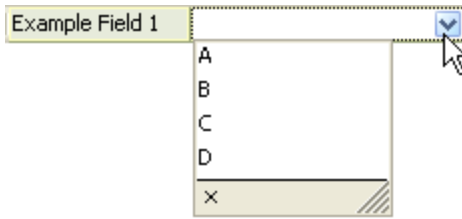
You can extend any baseline implementation in this way, by keeping the default inheritance statement and then adding your own custom code to the family rule project. Depending on how many functions are defined in the base class, you might choose to extend all of them or only some of them. Any function that is not extended within the family or field class itself will continue to use the baseline implementation.

Note that when you customize families and fields in this way, the family or field will *always* inherit the baseline customization, even if it changes in a future release. The customization itself, however, will *never* be modified by an upgrade.

For instance, continuing from the same example, assume that in a subsequent release, Meridium APM adds the values **X**, **Y**, and **Z** to the baseline implementation. Upgrading to the new version would affect only the *baselin*e Rules Library project. *The family rule project would not be modified*. In this case, because the family customization *inherits* the baseline implementation and then *adds to it*, the result would be a drop-down list that contains the values **A**, **B**, **X**, **Y**, **Z**, **C**, **D**. In other words, you would not need to make any changes to the family customization in order to apply the added baseline functionality.

> Note: This example illustrates a case where it would be desirable to inherit the updated baseline rule after an upgrade. Note, however, that while an upgrade will never modify your custom code on a syntactical level, changes to baseline rules could produce changes in your custom rules that are not desirable on a logical level. Depending on the degree to which baseline rules have changed, an upgrade of a database where this type of customization exists could cause families or fields to function improperly.

Overriding a Single Baseline Rule in a Class

The Meridium APM mechanism for storing baseline rules gives you the option of overriding some parts of the baseline implementation for a family or field but keeping others. To do so, for a given family or field class, you would *not change* the default Inherits statement. Then, within the family or field class itself, you would override the functions that you do not want to use.

For instance, consider the [example that we used to illustrate how baseline rules can be extended](). Instead, consider the following example, where the following code will override the default **PopulatePickList** function rather than *extending* it.

```vbnet
Option Strict On
Option Explicit On

Imports System
Imports Meridium.Core.DataManager
Imports Meridium.Core.DataManager.Customization
Imports Meridium.Core.Uom
Imports Meridium.Core.Internals
Imports Meridium.Core.Metadata

<MetadataField("EXAMP_FAMIL_EXAMP_FIELD_1_CHR")> _
Public Class EXAMP_FAMIL_EXAMP_FIELD_1_CHR
    Inherits Baseline.Example_Family.EXAMP_FAMIL_EXAMP_FIELD_1_CHR

#Region "Automatically generated code, do not modify!"
    Public SubNew(ByVal record As DataRecord, ByVal field As DataField)
        MyBase.New(record, field)
    End Sub
#End Region

    Public Overrides Sub PopulatePickList(ByVal pickList As DynamicPickList)
        pickList.Add("C")
        pickList.Add("D")
    End Sub
End Class
```

In this case, instead of *appending* the values **C** and **D** to the list constructed by the baseline code, this code specifies that the field customization should *override* the baseline customization. The result is that the list will contain *only* the values **C** and **D**.

This method of customization allows you to keep or discard baseline rules on a rule-by-rule basis. Keeping the default Inherits statement will cause the family or field to continue inheriting any function that is defined in the base class and not explicitly overridden within the family or field class itself. In addition, any functions that are not overridden within the family or field class itself and are modified in future versions of Meridium APM *will be applied* to the family or field after an upgrade. Functions that

have been overridden, however, will not be affected by changes delivered to the baseline Rules Library project in an upgrade.

Overriding All the Baseline Rules for a Class

By default, each family and field in the baseline Meridium APM database has a class that inherits functionality from a corresponding baseline class in the associated Rules Library project. If more than one rule is defined in the baseline Rules Library class, the Inherits statement will cause a family or field to inherit ALL of those rules. You may wish to keep the baseline inheritance and override specific functions within the family or field class. Alternatively, you can change the Inherits statement to override ALL of the baseline functionality.

> ⚠ **IMPORTANT:** If you choose to override the baseline rule by changing the family or field inheritance, in future upgrades of Meridium APM, your customization will not be affected by changes that are delivered to baseline rules.

To override ALL the baseline rules for a given family or field, simply modify the default Inherits statement to make the class inherit from a *different* class. This could be:

- **A class that defines *different* behavior.** This might be a class in the Client area of the Rules Library, a class in the Meridium APM area if the Rules Library, or a class defined in another family rule project. In this case, the behaviors defined in the base class would be applied *in place* of the behaviors defined in the baseline Rules Library class.

- **A class that defines no behavior.** Use this option if you want to define all the custom rule code within the family or field class itself. You can inherit the following classes, as appropriate for depending upon the type of code item you want to modify:
    - Meridium.Core.DataManager.Customization.EntityFamilyCustomization for entity family code items.
    - Meridium.Core.DataManager.Customization.RelationshipFamilyCustomization for relationship family code items.
    - Meridium.Core.DataManager.Customization.EntityFieldCustomization for entity family field code items.
    - Meridium.Core.DataManager.Customization.RelationshipFieldCustomization for relationship family field code items.

    Each of these classes has no behaviors associated with it, so only the rules that are defined directly within the family or field class itself will be applied.

> ⓘ **Hint:** You can use the fully qualified names in this list, or you can add a reference to the Meridium.Core.DataManager project and then use only the class name in the Inherits statement.

What is a Macro?

Business rules that reside within families and fields are executed when certain actions (i.e., create, update, or delete) are performed on records. For example, a **BeforeDelete** rule might exist on a family to prohibit the deletion of records when certain values exist in certain fields. The **BeforeDelete** rule will be executed automatically when a user initiates a Delete action, before the record is actually deleted.

Macros are rules that can be invoked *outside of* standard record functions via URLs to support workflows and invoke actions independently of creating, updating, or deleting records.

Macros are stored in the Rules Library. To create a macro, you create a Rules Library project and then write the rules that you want the macro to execute. After the macro exists, it can be invoked via a Meridium APM URL. Where you put the URL in the Meridium APM Framework application (e.g., on a Home Page or in Query results) will depend upon the workflow within which you want to execute logic defined in the macro.

A macro can be any static public method and can perform any function you like. Macros are most useful for facilitating workflows or for invoking custom functionality from the Home Page, Query Results, or a link on the **Associated Pages** menu. For example, you might want to develop a macro that displays a dialog box that prompts a user to make a choice and then performs an action based upon the choice that is made. Macros add extensibility and flexibility to the Meridium APM baseline features.

> Note: Because a macro can be any static public method defined in a Rules Library project, a macro can also be called from a family-level or field-level business rule by referencing the project and then calling the method. The main benefit of macros, however, is that they can be called *outside of* family-level and field-level rules to facilitate *any* workflow at the point-of-access that you choose and are not limited to being invoked by standard record functions.

Creating a Macro

A macro is any public static method defined in a Rules Library project. After you have created a macro, you will need to construct a URL to invoke it from the desired location (e.g., Home Page, query results, link on the **Associated Pages** menu).

To create a macro:

1. In the Configuration Manager, [create a Rules Library project](#).

2. Within the project, create a class, and define a public static method.

3. Within the method, write the desired code to define the functionality of the macro.

4. Compile the project.

   > ⓘ **Hint:** You can compile the project by clicking **Build [Project]** on the **Build** menu or by exiting the **Meridium APM Rules Editor** and compiling the project via the Rules Library.

5. Construct a URL to invoke the macro from the desired location. For example, you might want to:

   - Add a URL to a Home Page.
   - Add a URL to query results.
   - Create a link on an **Associated Pages** menu.

Using Sample Macros

The Meridium APM Rules Library includes sample macros that you can use as examples of macros that you can build in your system. Meridium APM provides three samples in the **Rules Library\Meridium\Samples\MacroSamples** project, each defined within a separate class:

- **CustomUI:** Provides an example of a macro that displays a simple user interface.
- **ShowMessageCreateEntity:** Provides an example of a macro that displays a message dialog box requesting a user response and then creates a new record.
- **UserDrivenNavigation:** Provides an example of a macro that displays a message box that provides the user with a choice of destinations. The screen that appears as a result of the macro will be determined by the user's response.

Each sample macro contains comments that explain how the macro works. The comments near the beginning of each class include the URL that will invoke the macro. To see how the macro works, copy the URL, paste it into a section on a Meridium APM Home Page, and then click the link to see what happens. The function of the macro should be self-explanatory after you have reviewed the comments.

> ⓘ **Note:** The **ShowMessageCreateEntity** macro contains a prompt that accepts the family ID of the family in which you want to create a new record. By default, the URL included in the comments uses the **Asset** family. If the **Asset** family does not exist in your database, you will need to modify this parameter for the URL to work. To do so, simply replace the text **Asset** with the ID of a family that exists in your database.

Because the sample macros are stored in the **Meridium APM** folder in the Rules Library, they cannot be modified. If you want to use any of the samples as the basis for creating your own macros, copy the Meridium APM code, paste it into a project in the **Client** folder, and then modify the code as desired. Keep in mind that to invoke your custom code you will need to modify the URL that is provided in the comments. For details on constructing URLs to invoke macros, see the URL Manager Help.

> ⓘ **Note:** The sample macros are provided simply to serve as examples of the functionality that can be implemented through macros. This documentation does not provide instructions on modifying the sample macros to customize their function. Creating and modifying macros requires knowledge of rule code that exceeds the scope of this documentation.

Parameters for the Macros URL

The URL to invoke a macro, **meridium://Registry/Macros**, accepts the parameters listed in the following table. Note that a link constructed using the URL path and no parameters will not be functional and will cause Meridium APM to display an error.

Parameter Name	Description	Accepted Value(s)	Notes
Macro	Specifies the name of the macro that you want to execute.	The name of a public shared method in the specified Rules Library project.	This parameter is required. You can specify the fully qualified method name or the name of the method itself. For more information, see the examples.
p0, p1, p2, etc.	Specifies values for parameters defined in the macro.	Any value that you want to pass in to a defined macro parameter.	Parameter values can be specified whenever one or more parameters have been defined in the macro. You can use the same syntax that you use for passing prompt values in to queries.
Project	Specifies the name of the Rules Library project in which the macro exists.	The name of a Rules Library project in which the macro has been defined.	This parameter is required.

Examples of the Macros URL

- meridium://Registry/Macros**?project=Rules Library\Client\Macros\Email&macro=SendEmail**

  Executes the SendEmail macro, which is stored in the project \\Client\Macros\Email in the Meridium APM Rules Library.

- meridium://Registry/Macros?project=Rules Library\Client\Macros\Email&macro=**Library.Client.Email.EmailClass.SendEmail**

  Executes the same macro as the first example but uses the fully qualified name rather than the method name.

- meridium://Registry/Macros?project=Rules Library\Client\Macros\Email&macro=**PromptUser&p0=Would you like to send an e-mail?**

  Executes the PromptUser macro, which is stored in the project \\Client\Macros\Email in the Meridium APM Rules Library and passes in the parameter value *Would you like to send an e-mail?* This example assumes that one parameter has been defined in the macro.

About Compiling Rules

For rules to work properly, they must be built, or *compiled*, against the version of the Meridium APM application that you are currently running. You will need to compile rules at various points throughout your installation, set-up, and configuration processes, including:

- After you upgrade the Meridium APM Database. For details on upgrading the database, see the Meridium Installation, Upgrade, and System Administration Help.
- Whenever you activate a license.
- Whenever you modify rules on a family, for a field, or within the Rules Library.

You have various options for how to compile the database. You can compile rules:

- Via the Meridium APM Rules Editor (VSTA).
- For individual entity and relationship families. This will also compile the rules for all fields within the family that you select to compile.
- For projects within the Rules Library or entire folders within the Rules Library.

In addition, you have the option of compiling the entire database. This option allows you to compile all the rules for all entity families, relationship families, and Rules Library projects in a single step.

Compiling Family Rules

After you make changes to entity or relationship family rules, you must compile the family for the changes to be applied. The following instructions provide details on compiling the rules for one or more families. Compiling families is necessary *only* if the family rule project exists. Families cannot be compiled until the family rule project has been created.

> 📄 **Note:** To compile an entire data source, you must compile all entity family rules, all relationship family rules, and all projects in the Rules Library. You can use the following instructions to compile rules for entity and relationship families.

To compile the rules for a family:

1. In the Configuration Manager, in the **Entity Family/Relationship Family** pane:

   - Click the **Entity Family** tab if you want to compile entity families.

   -or-

   - Click the **Relationship Family** tab if you want to compile relationship families.

2. Select the family whose rules you want to compile.

   > ⓘ **Hint:** To compile the rules for ALL families, select the **Root** family and then choose the **Compile Sub-Families** option on the **Family Compile** dialog box (see step 6).

3. In the **Tasks** section on the right side of the window, click the **Compile Family** link.

   The **Family Compile** dialog box appears.

4. If you want to compile all dependencies automatically, select the **Auto-Compile Dependencies** check box.

   > 📄 **Note:** If you are compiling a family that references one or more Rules Library projects, when you select this option, the referenced rule projects will be compiled as well.

5. If you want to compile all the subfamilies of the selected family, select the **Compile Sub-Families** check box. Note that this option is selected by default. You can clear this check box if you want to compile the selected family *only*.

6. Click the **Compile** button.

   The rules for the families that you selected are compiled.

   Note that:

   - If you select a family for which no rules exists, the message **No projects were selected** will be displayed at the top of the **Family Compile** dialog box.

- The compile log, which appears in the main display area of the **Family Compile** dialog box, reports only on families that were actually compiled. For example, if you select a family that the Meridium APM system determines does not need to be compiled, that family will be omitted from the log.

- The name displayed in the compile log is the *project* name, not the family name.

- You can save the compile log by selecting all the text that is displayed, pressing **Ctrl+C** to copy it, and then pasting it into another application, such as Notepad.

7. Click the **Close** button to close the **Family Compile** dialog box.

Compiling Rule Projects

To compile a single rule project:

1. In the Configuration Manager, in the Rules Library folder structure, select the folder that contains the project you want to compile.

2. In the pane on the right, select the project that you want to compile.

3. On the toolbar, click the **Compile Project** button 📇.

   The **Project Compilation** dialog box appears.
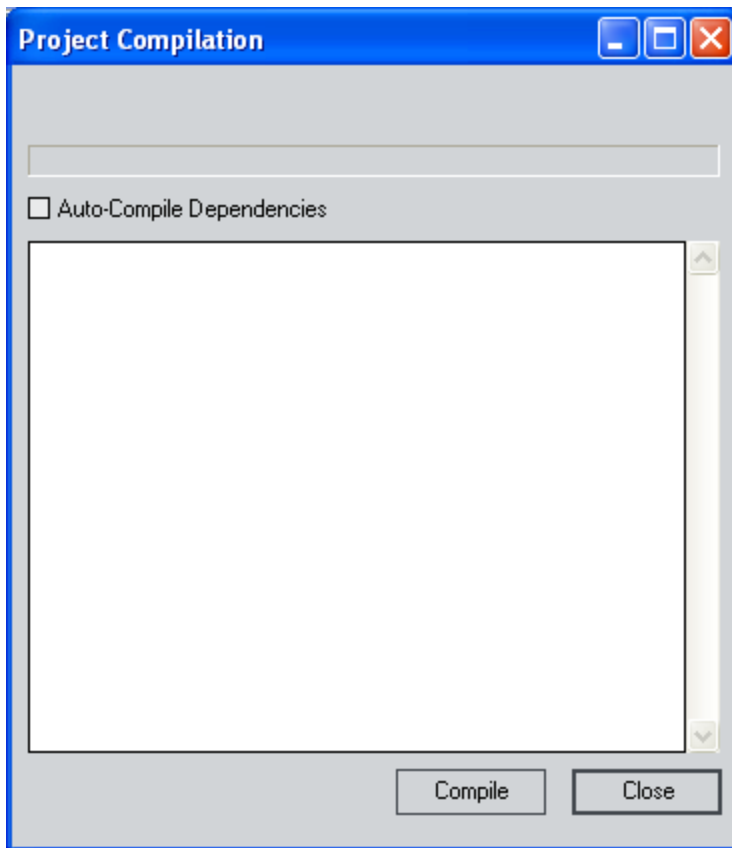
4. If you also want to compile dependencies, select the **Auto-Compile Dependencies** check box.

5. Click the **Compile** button.

   The system compiles the project and, if you specified to do so, any dependencies. The progress of the compilation is displayed on the **Project Compilation** dialog box.

6. When the project is finished compiling, click the **Close** button to close the **Project Compilation** dialog box.

Compiling Entire Folders

To compile ALL the projects within a given folder:

1. In the Configuration Manager, in the Rules Library folder structure, select the folder that you want to compile.

2. On the toolbar, click **Compile Folder** 📅.

   A prompt appears, asking if you also want to compile projects in the folder's sub-folders.

3. Click the **Yes** or **No** button, as desired.

   The **Project Compilation** dialog box appears.
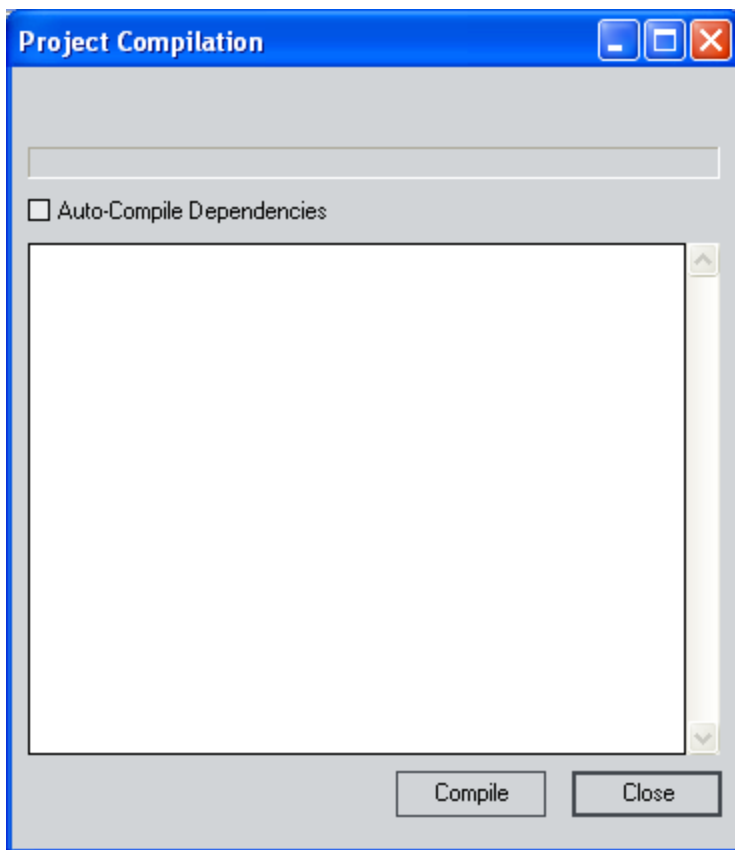
4. If you also want to compile dependencies, select the **Auto-Compile Dependencies** check box.

5. Click the **Compile** button.

   The system compiles the projects and, if you specified to do so, any dependencies. The progress of the compilation is displayed on the **Project Compilation** dialog box.

6. When the projects are finished compiling, click the **Close** button to close the **Project Compilation** dialog box.

Compiling the Entire Database

The following instructions provide details on compiling the entire database, including all entity and relationship family rule projects and all Rules Library projects.

To compile the entire database:

1. In the Configuration Manager, on the main menu, click **Tools**, and then click **Compile All Rules**.

   The **Project Compilation** dialog box appears.
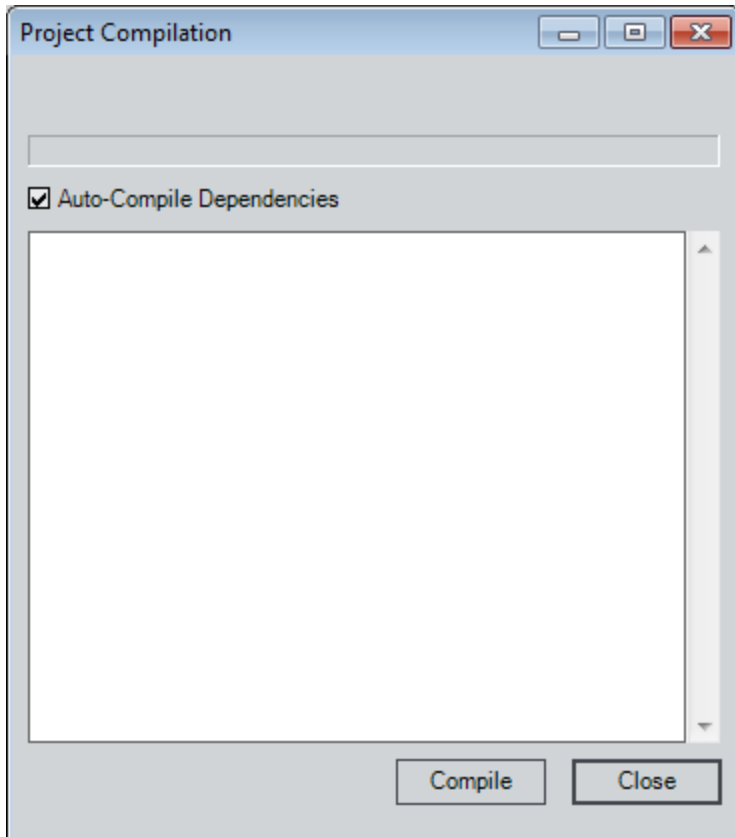
2. Accept the default selection for the **Auto-Compile Dependencies** check box (selected).

3. Click the **Compile** button.

   The compilation process begins. Rules Library project are compiled first, followed by entity families, and finally relationship families. The progress of the compilation is shown on the **Project Compilation** dialog box, as shown in the following image.
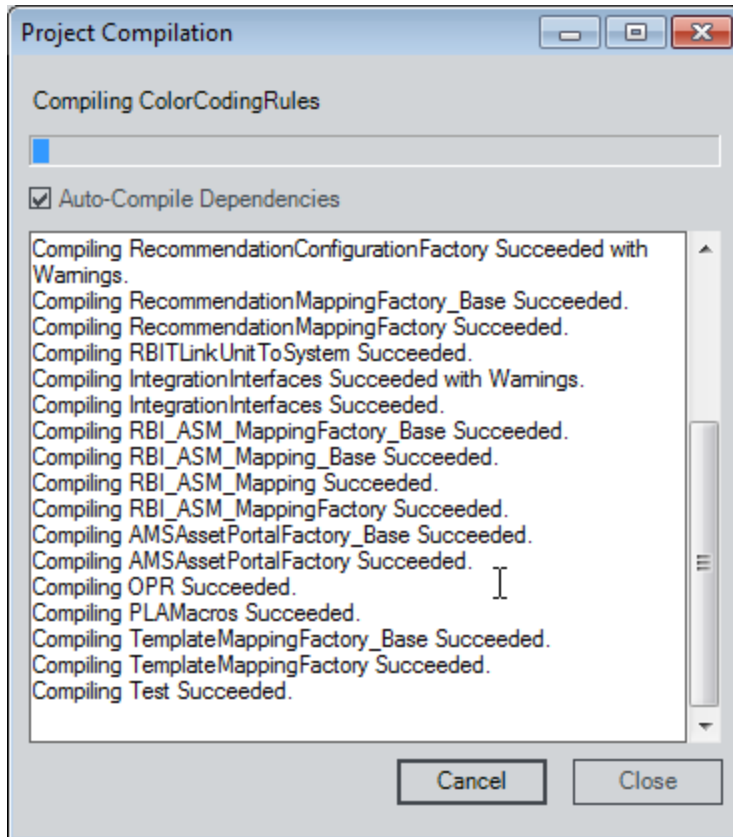
If you need to terminate the process, you can click **Cancel**. Otherwise, after the compilation process is complete, the **Cancel** button will become labeled **Compile** again, as shown in the following image.
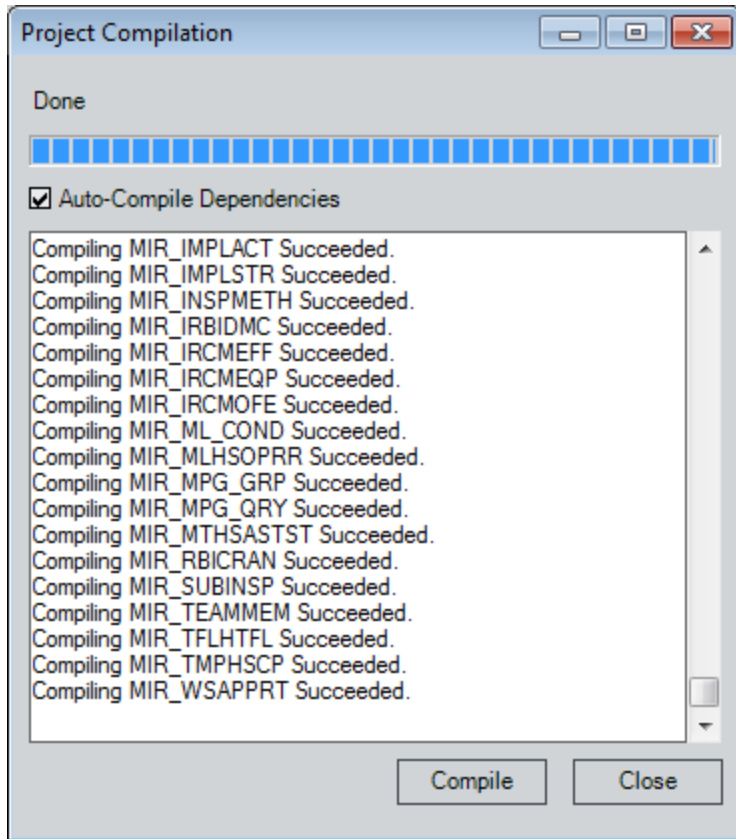
4. Click the Close button.

The **Project Compilation** dialog box closes.

About Content Validation

The Meridium APM product includes baseline content that is owned and controlled by Meridium, Inc. Customers can use the Meridium APM content as a starting point for developing their own content, but the baseline Meridium APM content should not be modified. Protecting the integrity of the baseline Meridium APM content is essential for supporting the Meridium APM database upgrade process and for ensuring that the Meridium APM product will continue to function properly.

The Meridium APM product enforces restrictions on modifying baseline content. For example, if you attempt to modify a baseline Rules Library project in the Meridium APM Rules Editor, you will not be able to do so. As long as these restrictions are in place, the baseline Meridium APM content will be protected. Through alternate and unsupported workflows, however, it may be possible to modify the baseline Meridium APM content. *If this occurs, you should restore the baseline content immediately.*

To help you ensure that your database does not contain any invalid baseline content and to help you detect problems as soon as they occur, Meridium APM provides the Content Validation tool. Using this tool, you can scan your database for invalid baseline content. Any invalid content that is detected can be exported, after which, Meridium APM Inc. can assist you in restoring the content to a valid baseline state.

The following content is protected by the Meridium APM Content Validation mechanism:

- Catalog folders and items.
- Meridium APM Rules Library projects and code items.
- Families.
- Fields.
- Registered URLs.
- Dependencies used for license activation.

When Content Validation is enabled, modifications to and deletions from baseline content will be will be detected. We recommend that you enable Content Validation to ensure the integrity of your database.

Enabling Content Validation

When content validation is enabled, each time a user logs in to Meridium APM, a check will be performed to ensure that the baseline Meridium APM content is valid. If the content is not valid, the user will have the option of exporting the invalid content or ignoring the warning.

We strongly recommend that you enable content validation in your system. The sooner invalid content is detected, the easier it will be to restore it to a valid state.

To enable content validation:

1. In the Configuration Manager, on the main menu, click **Tools**, and then click **Manage Content Validation**.

   The **Content Validation Wizard** appears, displaying the **Content Validation Status** screen.
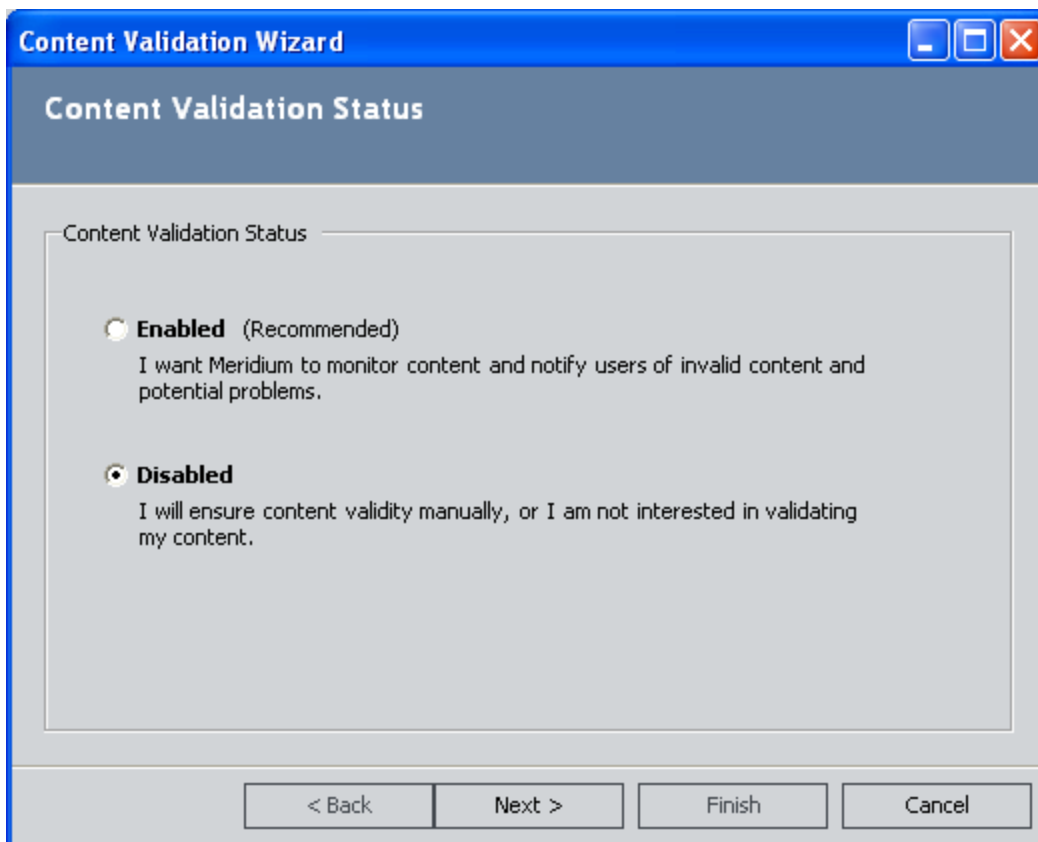
2. Select the **Enabled** option.

   ⓘ **Hint:** You do not need to change the selection on the **Content Validation Status** screen to proceed through the Content Validation Wizard. If Content Validation is currently enabled, you can accept the current selection and proceed

•

through the wizard simply to scan the database for invalid content and export a list of any invalid content that is detected.

3. Click the **Next** button.

    The Meridium APM system scans your database for invalid baseline content and displays the results on the **Content Summary** screen.

If invalid content was identified, it is displayed in the list on the **Content Summary** screen, and a message is displayed at the bottom of the screen indicating that invalid content was detected. Both modifications to and deletions from baseline content are considered invalid and will be included in the list.

4. If invalid content was detected, click the **Export** button to export the list of invalid content.

    The **Select export location** dialog box appears.

5. Choose a file name and location for the export file. Note that you can save the export file as a text file or an XML file.

6. Click the **Save** button.

   The **Select export location** dialog box closes, and a confirmation message appears, indicating that the export file was created successfully.

7. On the **Content Summary** screen of the **Content Validation Wizard**, click the **Finish** button.

   The **Content Validation Wizard** closes. Content validation is now enabled. The database will be scanned for invalid content each time a user logs in to Meridium APM.

Disabling Content Validation

To disable content validation on a database where it is enabled:

1. In the Configuration Manager, on the main menu, click **Tools**, and then click **Manage Content Validation**.

   The **Content Validation Wizard** appears, displaying the **Content Validation Status** screen.
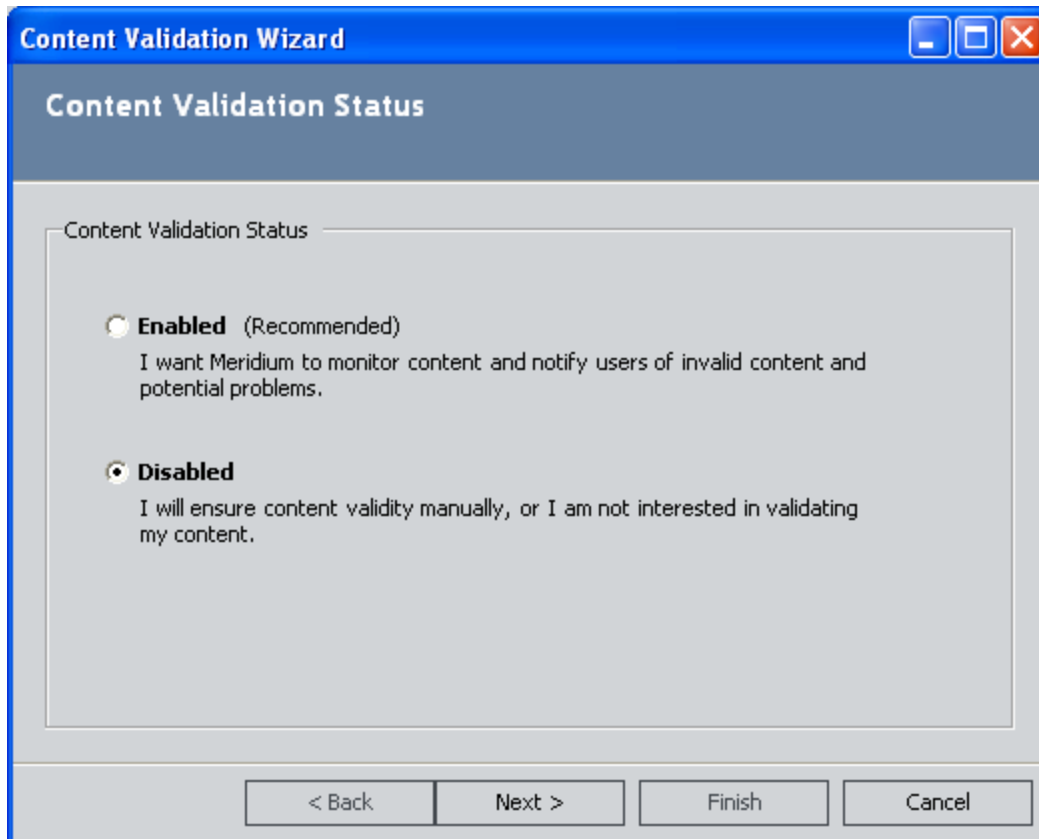
2. Select the **Disabled** option.

   > ⓘ **Note:** You do not need to change the selection on the **Content Validation Status** screen to proceed through the Content Validation Wizard. If Content Validation is currently disabled, you can accept the current selection and proceed through the wizard simply to scan the database for invalid content and export a list of any invalid content that is detected.

3. Click the **Next** button.

   The Meridium APM system scans your database for invalid baseline content and displays the results on the **Content Summary** screen.

•

If invalid content was identified, it is displayed in the list on the **Content Summary** screen and a message is displayed at the bottom of the screen indicating that invalid content was detected. Both modifications to and deletions from baseline content are considered invalid and will be included in the list.

4. If invalid content was detected, click the **Export** button to export the list of invalid content.

   The **Select export location** dialog box appears.

5. Choose a file name and location for the export file. Note that you can save the export file as a text file or an XML file.

6. Click the **Save** button.

   The **Select export location** dialog box closes, and a confirmation message appears, indicating that the export file was created successfully.

7. On the **Content Summary** screen of the **Content Validation Wizard**, click the **Finish** button.

   The Content Validation Wizard closes. Content validation is now disabled.

Overview of System Codes and Tables

System codes are frequently used to generate valid values lists in datasheets. By using System Codes, you can create a list of codes and then reference the list from any record or relationship field. These lists, when displayed in datasheets, can assist users in selecting the proper value for a given field.

Consider an Equipment Type field that has 20 values from which users can select. You can create a valid values rule to generate a list with the 20 values. If you use this Equipment Type field in many different equipment families, you can create a System Code Table and put the 20 Equipment Type codes in it instead of typing the valid values choices for each individual family. After creating the table and its codes, you can call the System Code Table from your valid values rule so that the lists are populated from the System Codes. If you need to change one of the Equipment Type codes, you can update it in the System Codes table, so that the next time the System Codes are called from a valid values rule, the changed code will appear without having to make any changes to the rule.

Additionally, you can choose to display System Codes in one of three ways: **Code Only**, **Description Only**, or **Code and Description**. When making these decisions, it is important to keep in mind the types of users who will be using the lists. For example, if you display only the codes, some users may not understand what each code represents. The most thorough selection would be **Code and Description** so that users can view both properties simultaneously.

It is important to keep in mind that the organization of the list becomes more important when there are numerous entries from which to choose. References categorize System Codes. If a System Code Table contains a large number of System Codes, it may be a good idea to organize them into groups. In doing so, you are narrowing down the end user's search. The end user could first choose from the references before making a final selection from the valid values list. For example, say that you have created a system code table that contained a System Code for every API code used in a field. With so many choices, it would be hard to decide. If you referenced the codes by breaking them up into different types, however, you could create two valid values lists: one in which the user selects the code type and another that references the first list and displays only those codes of the type selected in the first list. Also, System Codes can be sequenced to control their display order.

Accessing the System Codes and Tables Window

The **System Codes and Tables** window displays a list of all the System Code Tables that are defined for the database and the properties that are defined for each table.

To access the System Codes and Tables window:

- In the Configuration Manager, on the main menu, click **Tools**, and then click **System Codes and Tables**.

  The **System Codes and Tables** window appears.
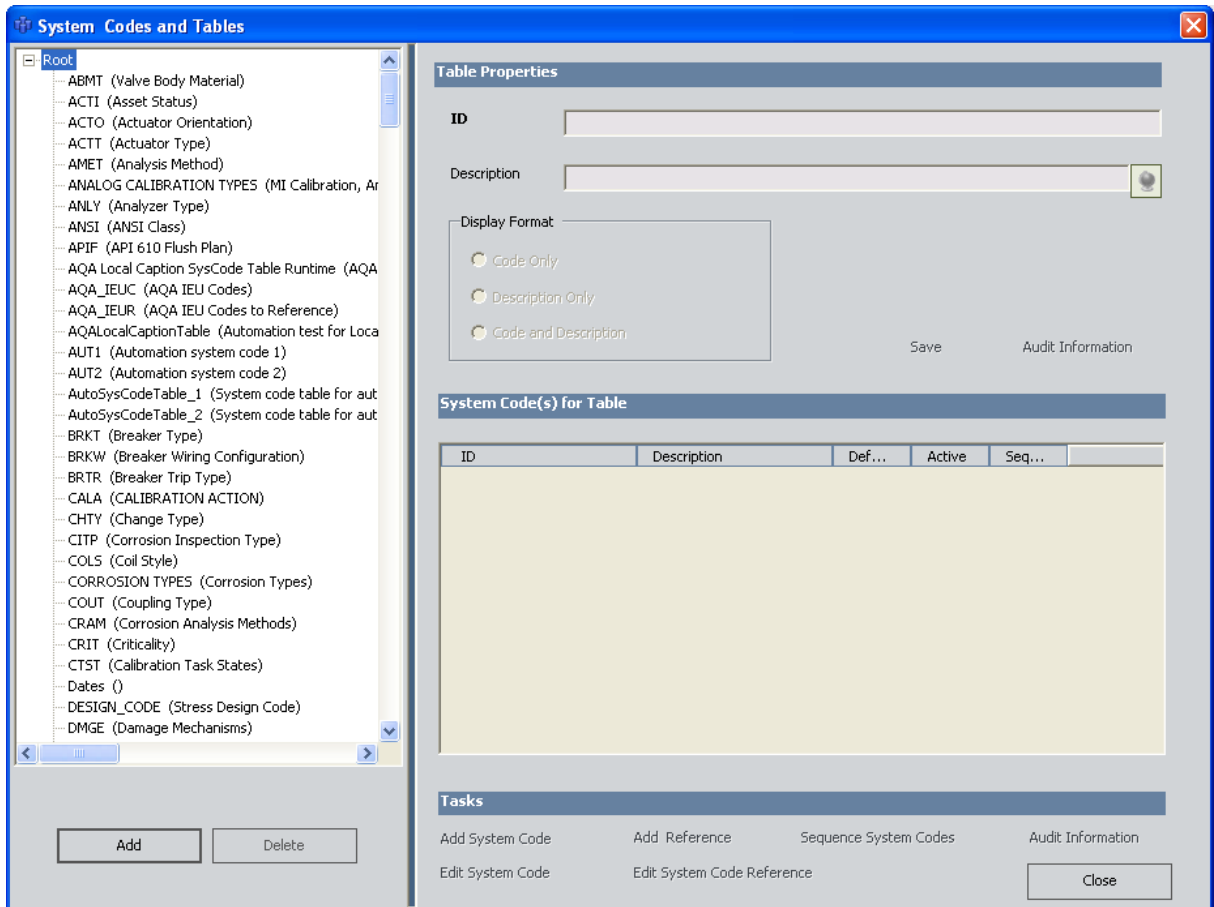
From the **System Codes and Tables** window, you can add a new system code table, view and modify the properties of an existing table, define System Codes for a table, or delete a system code table.

Adding a System Code Table

To add a System Code Table:

1. In the Configuration Manager, on the **System Codes and Tables** window, at the bottom of the list of System Code Tables, click the **Add** button.

   The **Add Table** dialog box appears.

2. In the **ID** text box, type the System Code Table ID. This ID is required and must be unique. The ID is limited to 50 characters.

3. In the**Description**text box, type a description for the table. This description should specify the table's purpose and which types of codes are present within the table. A description is not required but is recommended. The description is limited to 255 characters.

4. In the **Display Format** area, select a display format, which specifies how codes will be displayed. You can choose between **Code Only**(i.e., ID), **Description Only**, or **Code and Description**. Note that if you choose **Code Only** and your code is not very descriptive, it may be confusing to users.

5. Click the **Save** button.

   The new System Code Table is created and added to the list of all System Code Tables.

Viewing and Editing the Properties of System Code Tables

To view or modify the properties of system code tables:

1. In the Configuration Manager, on the **System Codes and Tables** window, in the list of System Code Tables, select the table that you want to modify.

   The table properties are displayed in the **Table Properties** section of the window.

2. If desired, modify the **ID**, **Description**, or **Display Format**.

3. Click the **Save** link to save your changes. Note that the **Save** link is enabled only after you modify at least one property of the System Code Table.

About System Codes

System codes are codes that are defined for a given System Code Table. Each table can have an unlimited number of codes defined for it. You can manage the codes associated with a System Code Table via the **System Codes and Tables** window. The System Codes that have been defined for a given table appear in the lower, right portion of the screen when you select a System Code Table.

The **Tasks** section displays various links, each of which corresponds to a function that you can perform related to System Codes.

Adding System Codes to a Table

To add a System Code to a System Code Table:

1. In the Configuration Manager, on the System Codes and Tables window, in the list of System Code Tables, select the table to which you want to add a System Code.

2. In the **Tasks**section, click the **Add System Code** link.

   The **Add System Code** dialog box appears.

3. In the **ID** text box, type the System Code ID. This field is required. The ID cannot exceed 50 characters in length and must be unique.

4. In the **Description**text box, type a description of the System Code. The description cannot exceed 255 characters in length.

5. Select the **Default** check box if you want this System Code to be the default code for the current System Code Table.

   > Note: Designating a default System Code has no effect on how that System Code will behave within Meridium APM.

6. If you do not want the System Code to be active, clear the **Active** check box.

   > Note: Only *active* System Codes will appear in valid values lists that are constructed from System Code Tables.

7. In the **Sequence** text box, type a value that indicates the order in which you want the code to be displayed in lists that are built from this System Code Table. This field is required and must be greater than 0 (zero). You should specify a value that is unique with respect to other System Codes within the table. The default value is one number greater than the current highest current sequence value for the table.

   > Hint: You can view and modify the sequence order of all System Codes in a given table by clicking the **Sequence System Codes** link in the **Tasks** section on the **System Codes and Tables** window.

8. Click the **Save** button.

   The new System Code is saved to the database.

Viewing and Modifying System Code Properties

To change or view a System Code properties:

1. In the Configuration Manager, on the **System Codes and Tables** window, in the list of System Code Tables, select the table that contains the System Code you want to view or modify.

2. In the **Tasks** section, click the **Edit System Code** link.

   The **View/Update System Code** dialog box appears.

3. Modify the properties as desired.

4. When you are finished making changes, click the **Save** button.

   Your changes are saved.

Changing the Sequence Order of System Codes

When you create a System Code, you can define the sequence number for that code, which defines the order of that System Code relative to other codes in the same System Code Table. If needed, you can modify the sequence order of the System Codes within a given System Code Table.

To change the sequence order of System Codes within a System Code Table:

1. In the Configuration Manager, on the System Codes and Tables window, in the list of System Code Tables, select the table that contains the System Code you want to view or modify.

2. In the **Tasks** section, click the **Sequence System Codes** link.

   The **Sequence Order** window appears, displaying a list of System Codes that exist for the selected table.

   In the list, the System Codes are arranged in order according to their current sequence numbers. By default, the **Manual** option is selected. Note that this option is always selected by default, regardless of the current sequence order of the System Codes.

3. Modify the order of the System Codes by:

   - Clicking the **Ascending** option to sort the System Codes in ascending alpha-numeric order.
   - Clicking the **Descending** option to sort the System Codes in descending alphanumeric order.
   - Selecting any System Code in the list, and clicking the up or down arrow to the right of the list to modify the sequence order of the System Codes. You can move System Codes up and down in the list using the arrow buttons regardless of which **Sort Order** option is selected.

4. After you have arranged the System Codes in the desired order, click the **Save** button.

   Your changes are saved, and the **Sequence** column on the **System Codes and Tables** window is updated to reflect the new sequence order for the System Code table.

Deleting System Codes from a Table

To delete a System Code:

1. In the Configuration Manager, on the **System Codes and Tables** window, in the list of System Code Tables, select the table from which you want to delete a System Code.

2. In the System Code Table tree structure, expand the selected table by clicking the + (plus sign).

3. In the list, locate the code that you want to delete.

4. Select the code, and click the **Delete** button, which appears at the bottom of the System Code Table list.

   A confirmation message appears, asking if you really want to delete the selected System Code.

5. Click the **Yes** button.

   The System Code is deleted.

What Are System Code References?

A System Code reference is a way of associating a System Code in one System Code Table with a System Code in another System Code table. For instance, consider an example where you have two System Code Tables: Equipment Type (EQPT) and Manufacturer (MFGR).

- The Equipment Type table contains the following values:
  - Compressor
  - Heat Exchanger
  - Motor
  - Pump
  - Tank
- The Manufacturer table contains the following values:
  - ACME
  - BURNS
  - SMITH

If the manufacturer *ACME* produces only motors and pumps, within the Manufacturer System Code Table, you could add a reference to the ACME System Code that references the Pump and Motor System Codes in the Equipment Type System Code Table. This would indicate that only the equipment types *Pump* and *Motor* are associated with the manufacturer *ACME*.

System Code references can be useful for developing Valid Values rules, where the values available in a given field are filtered based upon references to other System Codes. Keep in mind that the direction of the references will have an impact on how they can be used in rules. Before you begin creating references, you will want to think about how you plan to use them for developing rules so that you set them up properly. Throughout this documentation, we refer to the System Code that you select when you create the reference as the System Code *from* which a reference has been made *to* another System Code.

> **Note:** The direction of references cannot be determined by viewing them on the System Codes and Tables window. The direction is established at creation time, but when you view the references on the System Codes and Tables window, references appear to be bidirectional. It is important, therefore, that you take note of the direction when the reference is created.

Adding a System Code as a Reference

To add System Code(s) as a reference:

1. In the Configuration Manager, <u>on the **System Codes and Tables** window</u>, in the list of System Code Table tree, select the table that contains the System Code from which you want to add a reference.

2. In the **Tasks** section, click the **Add Reference** link.

    The **Add System Code as Reference** dialog box appears, displaying a hierarchical view of the System Code Tables and System Codes.
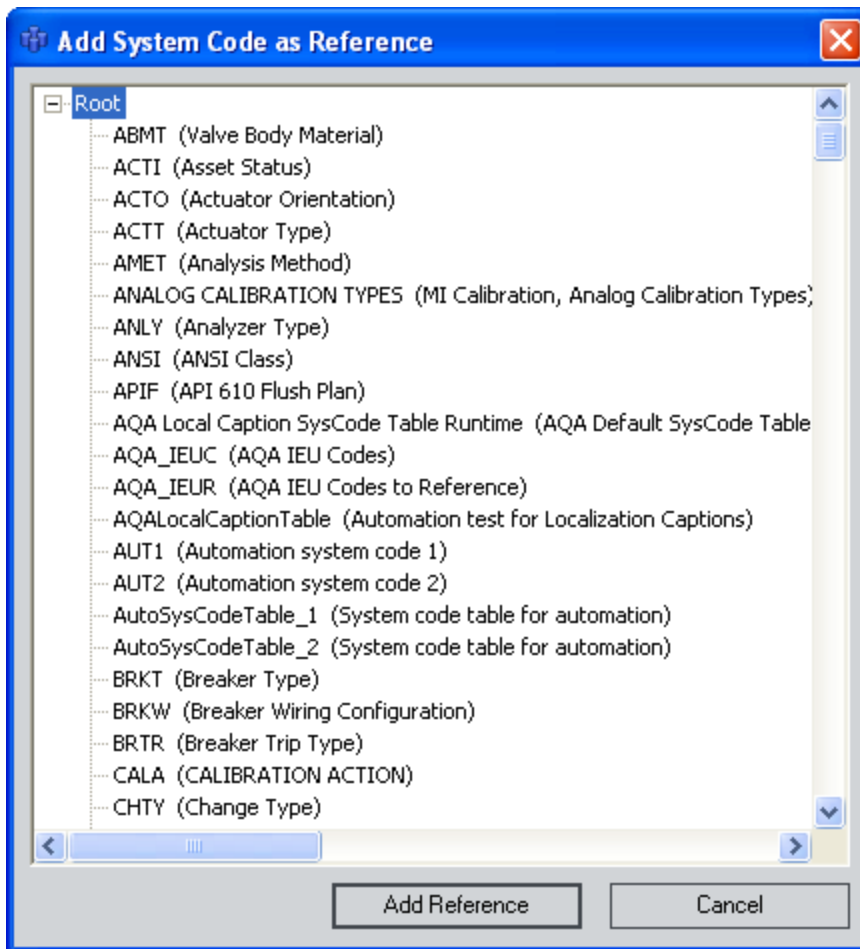
3. Select the System Codes that you want to reference from the System Code you selected in step 1.

    > ⓘ **Hint:** You can ALL the System Codes in a given table by selecting the System Code Table name.

4. Click the **Add Reference** button.

    The references are created.

Deleting System Code References

To delete a System Code reference:

1. In the Configuration Manager, on the **System Codes and Tables** window, in the list of System Code Tables, select the table from which you want to delete a System Code reference.

2. In the System Code Table tree structure, expand the selected table by clicking the + (plus sign).

3. Locate the code that contains the reference and click the + (plus sign) to expand it.

4. In the list, locate the reference that you want to delete.

5. Select the reference, and click the **Delete** button, which appears at the bottom of the System Code Table list.

   A confirmation message appears, asking if you really want to delete the selected System Code reference.

6. Click the **Yes** button.

   The reference is deleted.

Deleting a System Code Table

If you want to remove a System Code Table from the database permanently , you can delete it via the **System Codes and Tables** window.

To delete a System Code Table:

1. In the Configuration Manager, on the **System Codes and Tables** window, in the list of System Code Tables, select the table that you want to delete.

2. Click the **Delete** button.

   A confirmation message appears, asking if you really want to delete the table.

3. Click the **Yes** button to confirm the deletion.

About Units of Measure

Any numeric field in any family can have a unit of measure (UOM) associated with it. The UOM that is associated with a field helps characterize the values that are stored in that field. For example, a numeric field that is designated as having the UOM *Feet* will store values measured in feet. The UOM functionality in the Configuration Manager allows you to do three things:

- Set up the UOMs that will be available for associating with numeric fields. After you have defined units of measure, you can associate them with numeric family fields to characterize the type of data that is stored in the fields.

- Define how one unit of measure will be converted to another unit of measure (e.g., how inches will be converted to centimeters).

- Create UOM Conversion Sets, which are groups of UOM conversions that determine how values will be displayed to a certain set of users. After you have configured your Conversion Sets, you can associate them with the users in your system.

Accessing the Units of Measure Window

The units of measure that you define for your system determine which UOMs will be available for [associating with family fields](). You can manage all the UOMs in your system via the **Units of Measure** window.

To access the Units of Measure window:

- In the Configuration Manager, on the main menu, click **Tools**, and then click **Units of Measure**.

  The **Units of Measure** window appears, displaying the units of measure that are currently defined for your system.

From the **Units of Measure** window, you can [add UOMs](), [edit UOMs](), [delete UOMs](), and [configure conversion settings for UOMs]().

Adding a Unit of Measure

To add new unit of measure to the system:

1. In the Configuration Manager, on the **Units of Measure** window, scroll down to the last row in the grid. This row is empty and can be used for entering a new UOM.

2. Navigate to the **ID** field in the blank row, and enter the ID for the new unit of measure.

3. In the **Default Caption** cell, type a label for the UOM. This is how the unit of measure will be labeled when you create a record for a family whose datasheet contains the UOM field.

4. If desired, manage translations for that string.

5. In the **Description** cell, type a description of the unit of measure.

6. In the **Category** cell, type the category that best defines the unit of measure.

7. Click the **Save** button.

   The new UOM is saved to the database.

Editing a Unit of Measure

To edit an existing unit of measure:

1. In the Configuration Manager, <u>on the **Units of Measure** window</u>, navigate to the row containing the UOM that you want to modify.

2. In the **ID** field, modify the ID as desired.

3. In the **Default Caption** cell, edit the caption if desired.

4. If you want to manage translations for that string, click the icon, and add or modify the translations via the **Localize Default Caption** dialog box.

5. In the **Description** cell, modify the description if desired.

6. In the **Category** cell, modify the category.

7. Click the **Save** button to save your changes.

Deleting a Unit of Measure

Deleting a UOM permanently deletes it from the database. Additionally, deleting a UOM that is being used by one or more fields will cause those fields to have no UOM defined.

To delete a unit of measure from the database:

1. In the Configuration Manager, on the **Units of Measure** window, click the row that contains the UOM that you want to delete. An arrow appears to indicate that the row has been selected.

2. On your keyboard, press the Delete key.

   A confirmation message appears, asking if you really want to delete the unit of measure.

3. Click the **Yes** button.

4. On the **Units of Measure** window, click the **Save** button to save your changes.

About UOM Conversions

UOM conversions let you specify how one unit of measure will be converted to another unit of measure. The conversions that you define will be used to convert values stored in one UOM to another UOM for display. For example, you might define a conversion for the UOM *inches* that will convert values stored in *inches* to *centimeters* for display.

After you have defined conversions for the UOMs in your system, you can create Conversion Sets, which are sets of conversions that will be used by specific users. For example, if your database is configured to store values in English Standard units and you have a group of users who would prefer to view those measurements in Metric units, you could create an English-to-Metric Conversion Set and assign it to those users. In this way, your database would store the English Standard values, and when the system was accessed by a Metric user, the English Standard values would be converted and displayed in Metric units to that user.

Meridium APM provides a set of default UOM conversions that are used by Thickness Monitoring. You will need to configure manually any other conversion that you want to use for creating Conversion Sets.

> **Note:** To define how one UOM will be converted to another, *both* UOMs must first exist in your system. This means that before you can define the conversions for a UOM, you must first have added to the list of UOMs all the units of measure to which that UOM will be converted. Note also that if you define a conversion from one UOM to another UOM, you should *also* define a conversion from the converted UOM *back to the original UOM*. For example, if you define for the **Inches** UOM a conversion to *centimeters*, then for the **Centimeters** UOM you should define a conversion back to *inches*.

Specifying Conversions Between Units of Measurement

> ℹ **Note:** In the following instructions, we use the term *From Value* to mean the UOM *from* which a value is being converted and *To Value* to mean the UOM *to* which a value is being converted.

To define conversions for UOMs:

1. In the Configuration Manager, on the [Units of Measure window](#), select the row containing the unit of measure for which you want to specify conversions (i.e., the From Value).

2. Click the **Conversions** link.

   The **Unit of Measure Conversions** dialog box appears.

3. In the **Destination UOM** cell of the first blank row, select the unit of measure that will serve as the destination of the conversion (i.e., the To Value).

   > ℹ **Note:** Only UOMs in the same Category as the source UOM will appear in the drop-down list.

4. In the **Base** field, type the value that should be added to the From Value before the **Numerator** and **Denominator** are applied. To specify that a value should be *subtracted* from the From Value, type a negative number. A **Base** value of **0** (zero) indicates the From Value will not be modified before the **Numerator** or **Denominator** are applied.

   For example, the equation to convert Celsius to Fahrenheit is **C = (F - 32) / 1.8**, where **32** is subtracted from the From Value before it is divided by **1.8**. In this case, the **Base** would be **-32**.

5. In the **Numerator** and **Denominator** fields, define the fraction by which the From Value must be multiplied to arrive at the To Value. For example:

   - The equation to convert Centimeters to Inches is **cm = in x 2.54**. In this case, the **Numerator** would be **2.54** and the **Denominator** would be **1**.

   - The equation to convert Inches to Centimeters is **in = cm / 2.54**. In this case, the **Denominator** would be **2.54** and the **Numerator** would be **1**. Alternatively, you could set the **Numerator** to **0.39** and the **Denominator** to **1**.

6. In the **Offset** field, type the value that should be added to the From Value after the **Numerator** and **Denominator** have been applied. To specify that a value should be *subtracted* from the From Value, type a negative number. An **Offset** value of **0** (zero) indicates the From Value will not be modified further after the **Numerator** or **Denominator** are applied.

For example, the equation to convert Fahrenheit to Celsius is **F = C × 1.8 + 32**, where **32** is added after the From Value is multiplied by **1.8**. In this case, the **Offset** would be **32**.

7. If desired, repeat steps 3-7 to define additional conversions for the selected UOM. Each time you type information in the empty row, a new row will be added so that you can define as many conversions as you need.

> ⓘ **Hint:** If you want to delete a conversion from the list, highlight the row that you want to delete, and press the Delete key on your keyboard. Like additions and modifications, deletions will not be saved until you click the **Save** button.

8. When you are finished defining conversions for the selected UOM, click the **Save** button.

The conversions are saved to the database.

What Are UOM Conversion Sets?

A *UOM Conversion Set* is simply a group of [UOM conversions](). Each Meridium Security User will have a UOM Conversion Set associated with it to determine how numeric values will be converted and displayed throughout the Meridium APM applications for that user. You can manage all the UOM Conversion Sets in your system on the [UOM Conversion Sets window](), where you can add, modify, and delete UOM Conversion Sets.
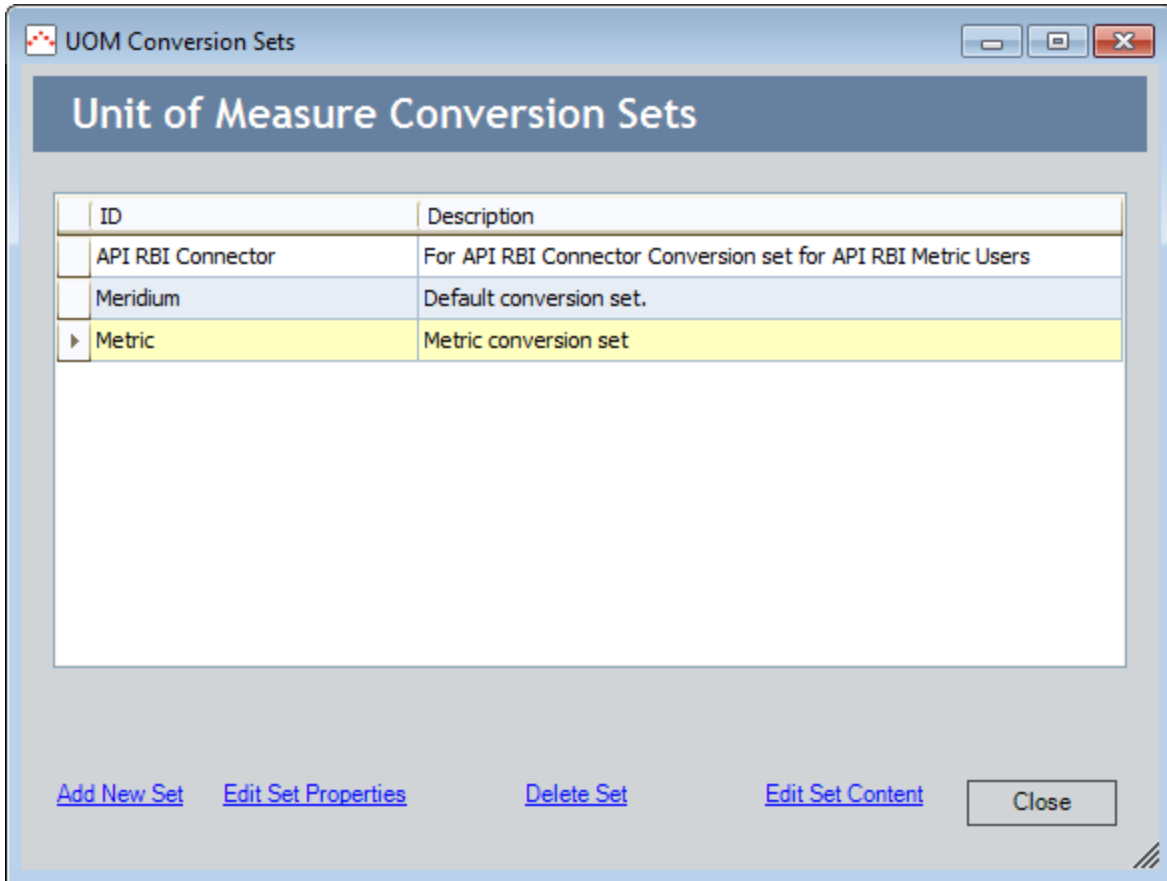
UOM Conversion Sets are useful for customizing the display of numeric values for the users in your system. Any numeric field can have a UOM defined for it to indicate the unit of measure for the *stored* value. The base (or stored) UOM, however, may not be appropriate for all users. For instance, some users may prefer to see a value that is stored in *inches* converted to and displayed as *centimeters*. Other users might prefer to see the same value converted to *millimeters*. You can allow for numeric values to be converted to different display values for different users by defining *UOM Conversion Sets*.
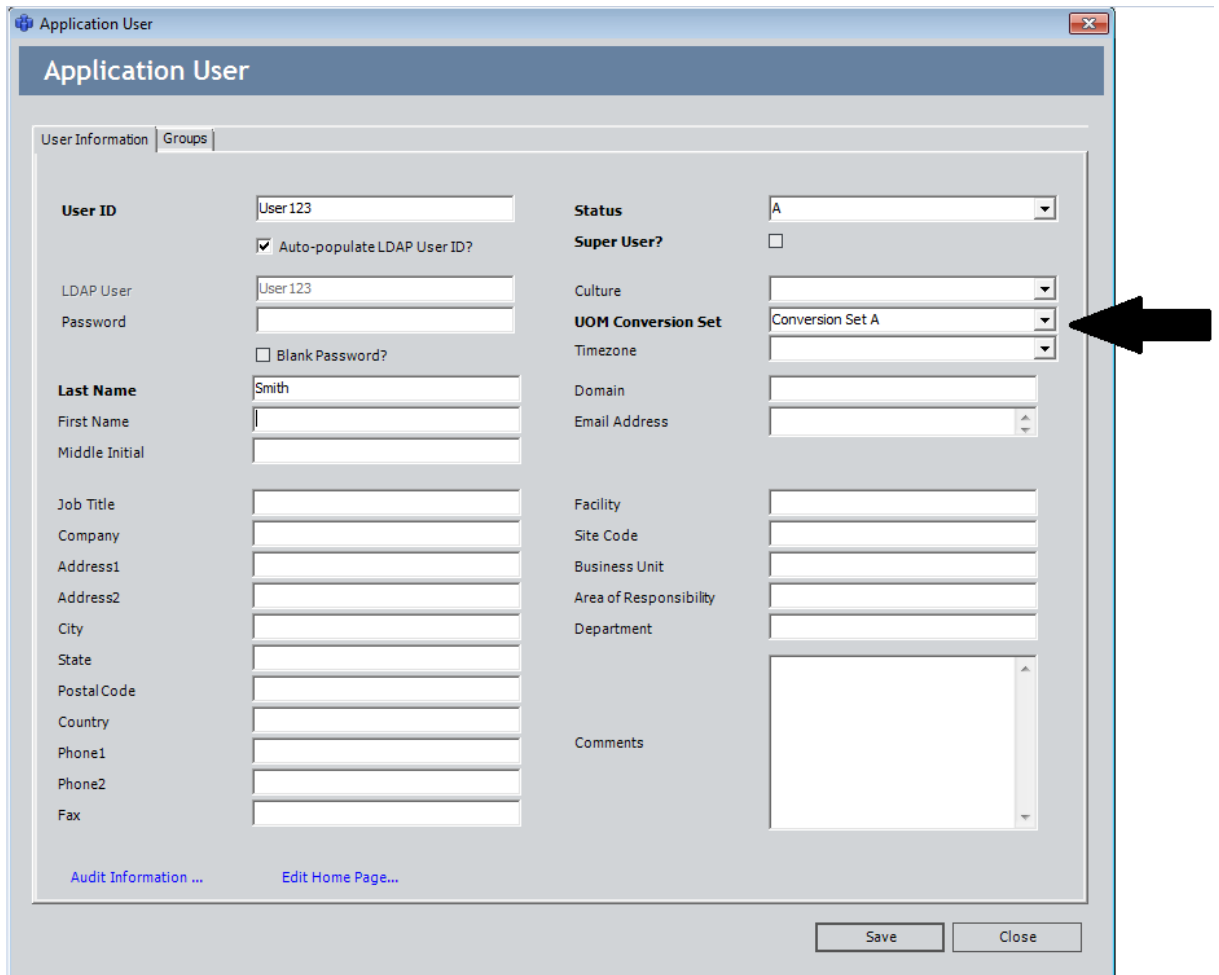
To take advantage of this functionality, you will need to complete the following steps:

1. Define the necessary base UOMs. For the example shown above, we would define *Inches*.

2. For each base UOM, define all required UOM conversions. For the example shown

above, we would create two UOM conversions, one for *Centimeters* and one for *Millimeters*.

3. For each type of user, create an appropriate UOM Conversion Set. For the example shown above, we would create two UOM Conversion Sets.

   - **Conversion Set A:** *From* Inches *To* Centimeters
   - **Conversion Set B:** *From* Inches *To* Millimeters

4. Associate each UOM Conversion Set with the Security User who will want to see the associated UOM conversions. For example, for any user who wanted to see values stored in inches displayed in centimeters, we would choose Conversion Set A, as shown in the following image.

Accessing the UOM Conversion Sets Window

To access the UOM Conversion Sets window:

- In the Configuration Manager, on the main menu, click **Tools**, and then click **Unit of Measure Conversion Sets**.

  The **UOM Conversion Sets** window appears.
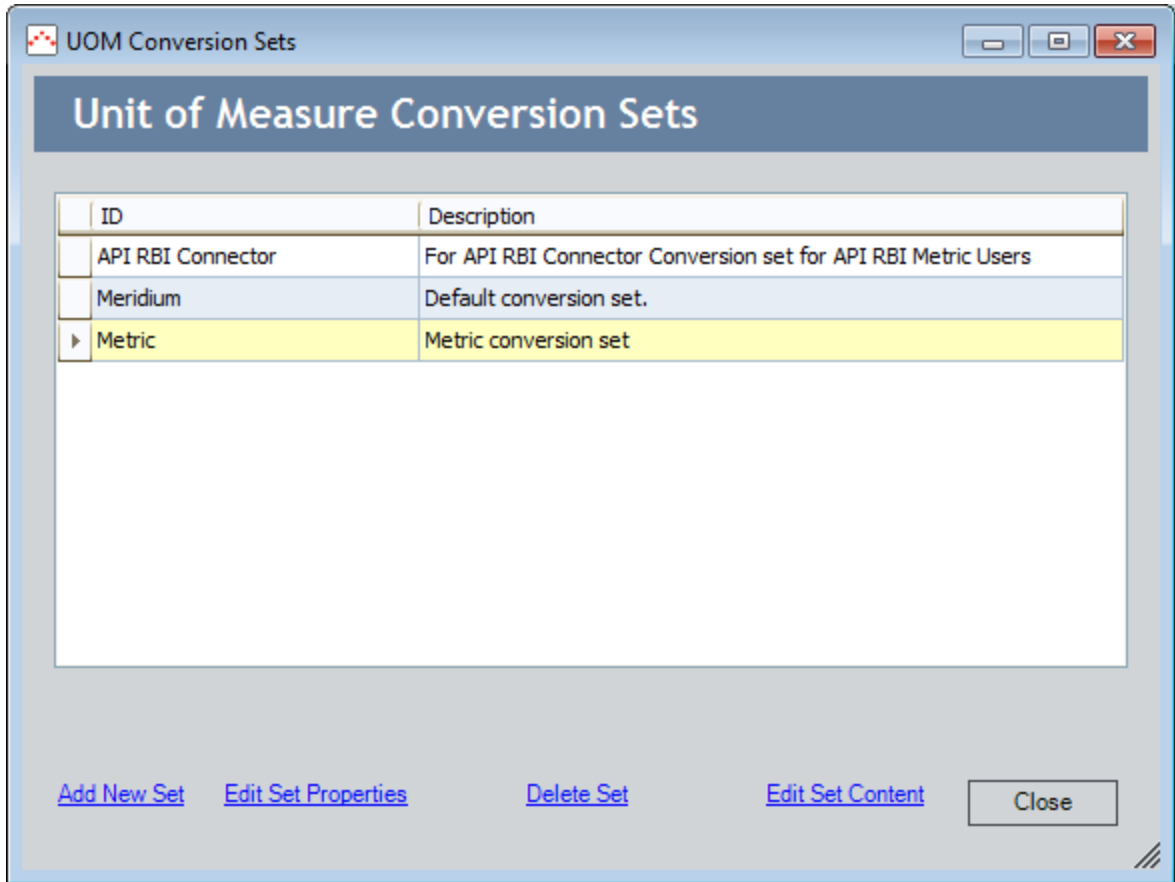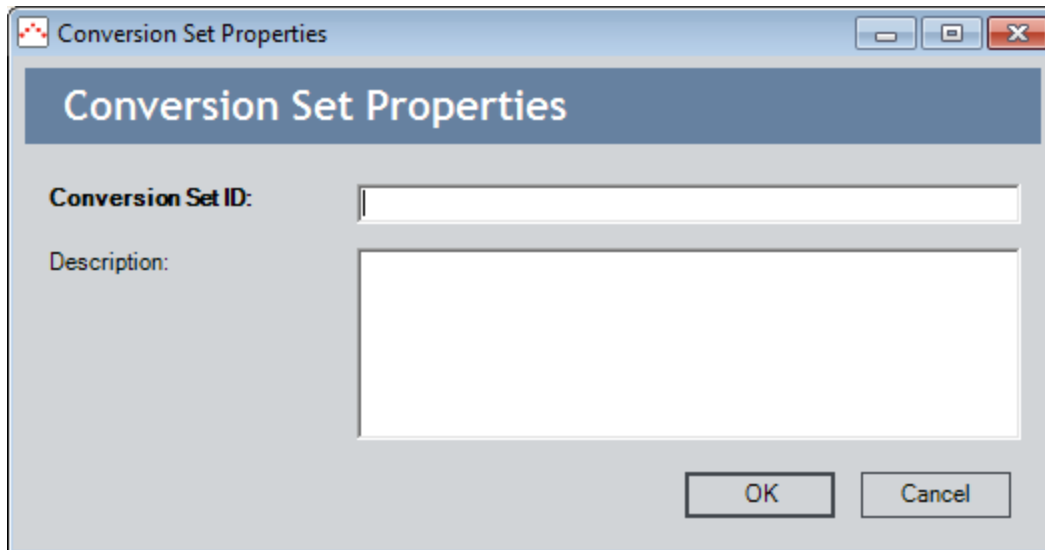
Creating a Conversion Set

To create a new UOM Conversion Set:

1. In the Configuration Manager, on the **UOM Conversion Sets** window, click the **Add New Set** link.

   The **Conversion Set Properties** dialog box appears.

2. In the **Conversion Set ID** text box, type an ID for the Conversion Set.

3. In the **Description** text box, type a description of the Conversion Set. The description is optional.

4. Click **OK**.

   The new Conversion Set appears in the list on the **UOM Conversion Sets** window. After you have created the Conversion Set, you can modify its content.

Modifying Conversion Set Properties

To modify the properties (i.e., the description) of an existing Conversion Set:

1. In the Configuration Manager, <u>on the **UOM Conversion Sets** window</u>, select the Conversion Set that you want to modify, and click **Edit Set Properties** link.

   The **Conversion Set Properties** dialog box appears, displaying the current ID and description of the Conversion Set.
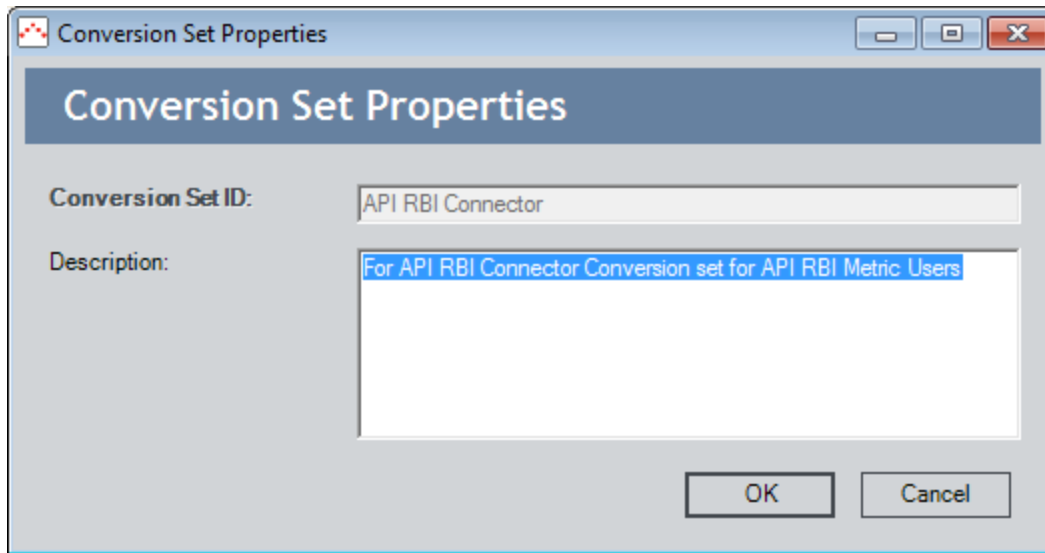
2. Modify the Description as desired.

   > 📄 **Note:** The **Conversion Set ID** field is disabled as the ID cannot be modified.

3. Click **OK.**

   Your changes are saved.

Modifying the Content of a Conversion Set

The content of a Conversion Set defines how values stored in one UOM will be converted for display in another UOM when a user uses that Conversion Set. You can modify the content of any user-defined Conversion Set. You cannot modify the content of the baseline Meridium APM Conversion Set. For each Conversion Set, you should define *at least* the following conversions, which are defined in the Meridium APM Conversion Set and are used by Thickness Monitoring:

From	To
Inches/Year	Mils/Year
Days	Months

> [i] **Note:** Values stored using UOMs that are not defined for a Conversion Set will be displayed using the *stored* UOM. This means that for a given Conversion Set, you need to define only the UOMs that should be displayed *differently* from the stored value.

To modify the content of a Conversion Set:

1. In the Configuration Manager, on the <u>UOM Conversion Sets window</u>, select the Conversion Set whose content you want to modify, and click the **Edit Set Content** link.

   The **Conversion Set Content** dialog box appears.
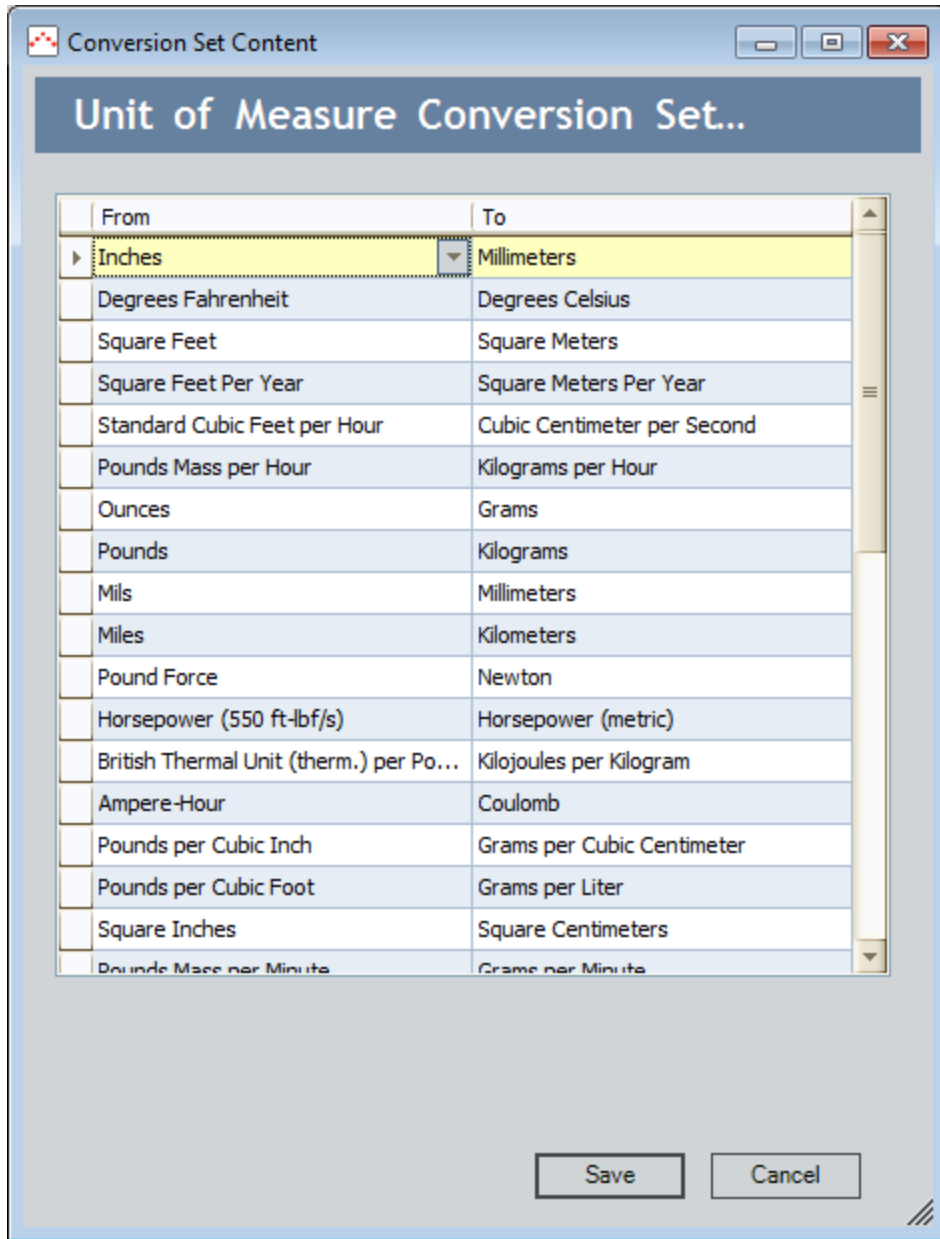
2. To add a conversion to the list:

- In the **From** column of the first empty row, select the UOM that defines the *stored* value. The **From** list displays the UOMs for which conversions have been defined and that have not yet been selected in this Conversion Set.

- In the **To** column of the same row, select the UOM to which values stored in the **From** UOM will be converted for *display*. This list contains only the UOMs for which conversions have been defined for the UOM selected in the **From** column. For example, if you have defined conversions for **Inches to Feet** and **Inches to Centimeters**, when you select **Inches** in the **From** column, the **To** column will list **Feet** and **Centimeters**.

•

3. To delete a conversion from the Conversion Set, select the row that you want to delete, and press the Delete key on your keyboard.

4. When you have finished defining the content of the Conversion Set, click the **Save** button.

   Your changes are saved to the database.

Deleting a Conversion Set

From the **UOM Conversion Set** window, you can delete any user-defined Conversion Set. You cannot delete the system-defined baseline Meridium APM Conversion Set.

> 🛈 **Note:** If you delete a UOM Conversion Set that is being used by one or more users, those user records will be modified to use the baseline Meridium APM Conversion Set.

To delete a conversion set from the database:

1. In the Configuration Manager, on the **UOM Conversion Sets** window, select the row containing the UOM Conversion Set that you want to delete.

2. Click the **Delete Set** link.

   A message appears, asking if you really want to delete the UOM Conversion Set.

3. Click the **Yes** button.

   The UOM Conversion Set is deleted from the database.

About Baseline UOM Conversion Sets

The following UOM Conversion Sets are provided with the baseline Meridium APM product:

- [Meridium](#)
- [Metric](#)

Any of these UOM Conversion Sets can be associated with any Security User. The *Meridium* [UOM Conversion Set](#) is associated with all new Security Users by default.

Meridium Conversion Set

The Meridium Conversion Set is included in the baseline product and contains the content listed in the following table.

> 🗋 **Note:** The Meridium Conversion Set it assigned to all new Security Users by default when they are first created. You can accept the default selection or choose a different UOM Conversion Set, as desired.

From	To
IN/YR (TM)	Mils/year
Days (TM)	Months

•

Metric Conversion Set

The *Metric* Conversion Set is provided in the baseline product and contains the content listed in the following table.

From	To
Inches	Millimeters
Degrees Fahrenheit	Degrees Celsius
Square Feet	Square Meters
Square Feet Per Year	Square Meters Per Year
Standard Cubic Feet per Hour	Cubic Centimeters per Second
Pounds Mass per Hour	Kilograms per Hour
Ounces	Grams
Pounds	Kilograms
Mils	Millimeters
Miles	Kilometers
Pound Force	Newton
Horsepower (550 ft-lbf/s)	Horsepower (metric)
British Thermal Unit (therm.) per Pound	Kilojoules per Kilogram
Ampere-Hour	Coulomb
Pounds per Cubic Inch	Grams per Cubic Centimeter
Pounds per Cubic Foot	Grams per Liter
Square Inches	Square Centimeters
Pounds Mass per Minute	Grams per Minute
Pounds/Sq Inch Gage	BAR(G)
Cubic Inches	Cubic Meters
Barrels	Liters
Gallons	Liters
IN/YR (TM)	Millimeters per Year
Pounds Force per Square Inch	Bar

From	To
Feet per Second	Meter per Second
Feet per Minute	Meter per Minute
Pounds Force - Foot	Newtons-Meter
Degrees Rankine	Kelvin
Standard Atmosphere	Bar
Pounds Force per Square Inch Absolute	Bar
Feet	Meters
Cubic Feet	Cubic Meters
Standard Cubic Feet per Minute	Liters per Minute
Inches per year	Millimeters per Year
Days (TM)	Months
Mils/year	Millimeters per Year

• Page 338 of 529

Overview of State Configuration

When a Meridium APM Framework user is working with records, they might need a way to indicate the status of a record and its associated information. For example, suppose the workflow for requesting maintenance work involves creating a Work Order record, assigning the record to the appropriate maintenance personnel, and then closing the record after the work is completed. In this case, it might be useful to show the status of the Work Order record as either *Created*, *Assigned*, or *Closed*. The record's *state* then serves as a visual indicator regarding the status of the record and the information that it contains.

Via the Configuration Manager application, you can define a *State Configuration* for a family so that Meridium APM Framework users can apply statuses to records in those families. A family's State Configuration consists of the combination of:

- *States* that identify the status of a record.
- *Operations* that define the available transitions between states.
- Rules that dictate a record's available states and the transitions that can be used to change the state.

You will define a family's State Configuration by accessing the **State Configuration** window for that family.

Meridium APM provides a baseline State Configuration for several Meridium APM families. You can define a State Configuration for any family of your choice.

After State Configuration is defined for a family, via the Meridium APM Framework application, you can also assign Security Users to State Configuration Roles and states to limit the users that are allowed to change a record's state.

Complete State Configuration Workflow

The following table outlines the basic workflow for defining State Configuration for a family and ensuring that the appropriate Security Users can transition records belonging to that family. Note that some of these tasks will need to be completed in the Configuration Manager, and others will need to be completed in the APM Framework.

Step	Application	Task
1	Configuration Manager	Identify the family for which you want to define State Configuration.
2	Configuration Manager	If desired, create State Configuration Roles that you want to assign to states that belong to that family's State Configuration.
3	Configuration Manager	Define the desired State Configuration for the family.
4	Configuration Manager	If desired, for the families for which you defined State Configuration, assign State Configuration Roles to states.
5	APM Framework	If you assigned State Configuration Roles to states, assign Security Users to those State Configuration Roles.
6	APM Framework	If you selected the **Require a specific user to be assigned to a state** check box for any state, assign Security Users to those states.

Example of State Configuration

Consider the following example of states and operations that you might configure for the Work Order family.

> ℹ **Note:** The Work Order family is not provided in the baseline Meridium APM data model. This example is provided to help explain states and operations for an item that has an inherent workflow: a work order.

Note that in the following image, a square represents a *state*, and a line represents an *operation*. The square at the beginning of an arrow represents a *predecessor state*, and a square to which the arrow points represents a *successor state*.

From this diagram, you can see that a Work Order record can be set to any of the following states:

- Proposed
- Rejected

- Approved
- Assigned
- In Progress
- Completed

The operations that cause a record to change from one state to another are:

- **Reject:** Changes the state from Proposed to Rejected.
- **Approve:** Changes the state from Proposed to Approved.
- **Assign:** Changes the state from Approved to Assigned.
- **Begin Work:** Changes the state from Assigned to In Progress.
- **Close:** Changes the state from In Progress to Completed.

Keep in mind that states are intended to provide a visual indicator of the status of a certain record and its associated information. In addition, operations help users determine what steps should be taken next. In most cases, the same user will not be performing all operations. Instead, one user will be responsible for one operation or a small set of operations, while another user will be responsible for another small set of operations. In this way, operations help guide users to perform the tasks for which they are responsible while prohibiting them from performing tasks that they should not participate in.

For example, based on the diagram shown above, for a Work Order record that is Approved, the only available operation to a Meridium APM Framework user would be *Assign*. This operation indicates that the record has been approved and is ready to be assigned to the appropriate user. Likewise, an Assigned record is ready to be implemented, which the operation Begin Work indicates. It is unlikely that the user assigning the work order would be the same who does the work.

As you define the State Configuration for a family, keep in mind the workflow that you want users to follow. Then, create states and operations that facilitate that workflow and decrease the possibility of user error.

About Baseline State Configurations

Meridium APM provides a baseline State Configuration for several families. If desired, you can extend a baseline State Configuration by adding new states and operations. You cannot, however, remove baseline states and operations that are flagged as Reserved in the **State Configuration** window. This limitation ensures that any baseline functionality that you can perform in the Meridium APM Framework application using states and operations does not get modified or removed. Note that if an operation is reserved, Meridium APM Framework users and Meridium APM Web Framework users will not be able to select that operation from the datasheet in the Record Manager.

> ⓘ **Hint:** You can modify or remove any baseline states or operations that are not flagged as Reserved.

About Accessing the State Configuration Window for a Family the First Time

The first time that you access the **State Configuration** window for a family:

- Three new fields will be created automatically for the selected family: State, Owner, and Last Entered Date. You can add these fields to any datasheet if you want Meridium APM Framework users to be able to see the values in places such as the Record Manager.

  > ⓘ **Hint:** If you add the fields to a datasheet, we recommend that you modify the datasheet captions. If you do not modify the captions, the default captions will be MI_SM_STATE_ID_C, MI_SM_STATE_OWNER_ID_C, and MI_SM_STATE_ENTERED_D.

- The State Configuration will be *enabled* if:
    - The family exists at the Root level of the family hierarchy.
    - The family is a subfamily but State Configuration is not defined for its parent family.

  You can disable the State Configuration manually.

- The State Configuration will be *disabled* if the family is a subfamily of a family for which State Configured IS defined. You can enable the State Configuration manually.

Accessing the State Configuration Window

To access the State Configuration window:

1. In the Configuration Manager, on the **Entity Family** tab, select the family for which you want to define State Configuration.

2. In the **Tasks** section, click the **Manage State Configuration** link.

   A message appears, indicating that since this is the first time that State Configuration has been invoked for this family, three new fields will be created automatically for the selected family: State, Owner, and Last Entered Date. The message asks if you want to continue managing State Configuration for the selected family.

3. Click the **Yes** button.

   A message appears, indicating that the fields were created successfully and that after you are finished defining the State Configuration, you must recreate the database views for the selected family and its subfamilies.

   > ⓘ **Hint:** Despite the information in this message, the Meridium APM system recreates the database views for the selected family automatically. You do not need to recreate the database views manually when you are finished defining State Configuration for a family.

4. Click **OK**.

   The **State Configuration** window appears.

Enabling or Disabling State Configuration

By default, the first time that you access the **State Configuration** window for a family, the State Configuration will be *enabled* if:

- The family exists at the Root level of the family hierarchy.
- The family is a subfamily but State Configuration is not defined for its parent family.

If desired, you can disable the State Configuration manually.

The State Configuration will be *disabled* if the family is a subfamily of a family for which State Configured IS defined. You can enable the State Configuration manually.

When State Configuration is enabled, the states and operations that are defined will be used on all new and existing records that exist in that family. This means that even if you choose not to display the states and operations on the datasheet, all new records will at least be assigned the initial state. If State Configuration is enabled and you choose to display the states and operations on the datasheet, Meridium APM Framework users who are assigned to the appropriate states will be able to transition records from one state to another via the datasheet.

To enable State Configuration when it is disabled:

1. In the Configuration Manager, access the **State Configuration** window for the family for which you want to enable State Configuration.

2. Select the **Enable State Functionality** check box.

3. Click the **Save** button.

   The State Configuration is enabled.

To disable State Configuration when it is enabled:

1. In the Configuration Manager, access the **State Configuration** window for the family for which you want to enable State Configuration.

2. Clear the **Enable State Functionality** check box.

3. Click the **Save** button.

   The State Configuration is disabled.

Showing or Hiding States and Operations on Datasheets

After State Configuration is enabled for a family, you can choose whether or not you want the states and operations to appear on the datasheets that are defined for the selected family. For example, in the following image, you can see on the Meridium General Recommendation datasheet that the state is *Created*, and the available operation is *Review*.
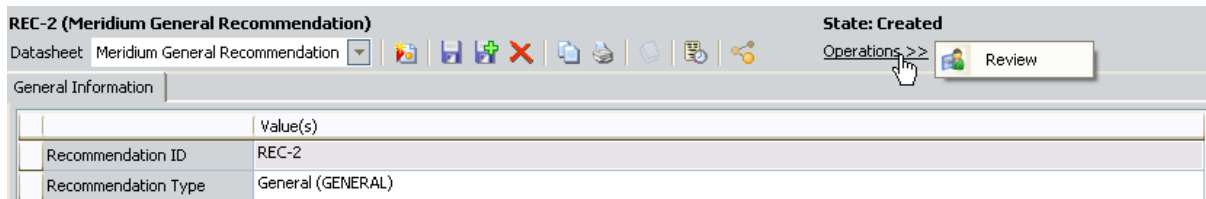
To hide states and operations on datasheets if they are shown:

1. In the Configuration Manager, access the **State Configuration** window for the family whose states and operations you want to hide on datasheets.

2. Clear the **Show States on Datasheets** check box.

3. Click the **Save** button.

   The states and operations will not be shown on datasheets defined for the selected family.

To show states and operations on datasheets if they are hidden:

1. In the Configuration Manager, access the **State Configuration** window for the family whose states and operations you want to hide on datasheets.

2. Select the **Show States on Datasheets** check box.

3. Click the **Save** button.

   The states and operations will be shown on datasheets defined for the selected family.

Adding New States

To add a new state to a State Configuration:

1. In the Configuration Manager, [access the **State Configuration** window](#) for the family to which you want to add a state.

2. If the **Enable State Functionality** check box is cleared, select it.

3. Below the **States** tab, click the **Add State** link.

   The state is added to the grid on the **States** tab, and the **ID** text box below the grid is populated automatically with the text **NewState_[n]**, where **n** is a digit one number higher than the number of current states defined for the family.

4. In the **ID** text box, delete the text that was generated automatically, and type an ID for the new state.

   > ⓘ **Note:** The **Caption** text box is populated automatically with the value that you typed in the **ID** text box. You can change the caption if desired. These instructions assume that you do not want to change the caption.

5. To the right of the **Caption** text box, if desired, click the 🌐 button to mange translations for that string.

6. If this state should be the initial state of all new records in this family, select the **Initial State** check box. Note that only one state in a family's State Configuration can be the initial state.

Adding New Operations

Note that before the State Configuration can be saved successfully, you must add at least two states and one operation that includes one of the states as a predecessor.

To add a new operation to a State Configuration:

1. In the Configuration Manager, access the **State Configuration** window for the family to which you want to add an operation.

2. If the **Enable State Functionality** check box is cleared, select it.

3. Click the **Operations** tab.

4. Below the **Operations** tab, click the **Add Operation** link.

   The operation is added to the grid on the **Operations** tab, and the **ID** text box below the grid is populated automatically with the text **NewOperation_1**.

   > ⓘ **Note:** These instructions assume that if this is not the first operation you are adding, you have added all others using the instructions provided in this documentation (i.e., you have changed the operation ID to something other than the default ID). In this case, the default ID will be **NewOperation_1**.

5. In the **ID** text box, delete the text that was generated automatically, and type an ID for the new operation.

   > ⓘ **Note:** The **Caption** text box is populated automatically with the value that you typed in the **ID** text box. You can change the caption if desired. These instructions assume that you do not want to change the caption.

6. To the right of the **Caption** text box, if desired, click the 🌐 button to manage translations for that string.

7. In the **Predecessor State** list, select the state that should serve as the predecessor state, or the state *from which* this operation will transition a record. For example, if a record can go from Created to Assigned via the Assign operation, to add the Assign operation, you would need to select the Created state as the predecessor state.

8. In the **Successor State** list, select the state that should serve as the successor state, or the state *to which* this operation will transition a record. For example, if a record can go from Created to Assigned via the Assign operation, to add the Assign operation, you would need to select the Assigned state as the successor state.

Defining Images for States and Operations

When defining a State Configuration for a family, after you add states and operations, you can associate an image with each state and operation. Images will be displayed in the Meridium APM Framework application in the following ways:

- **State image:** Displayed in the Meridium APM Framework application only if it is defined for a state that exists for the Recommendation family or one of its sub-families. In this case, the image will appear in the **Current State** column on the **Recommendation Management** page.
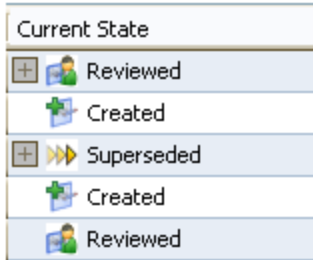
- **Operation image:** Displayed to the left of the operation when you are viewing the list of operations from a datasheet.
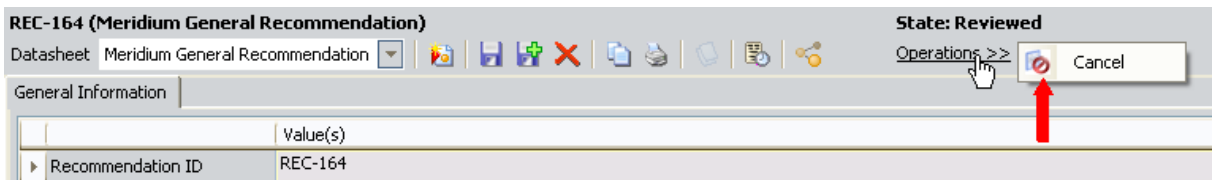
Note that to define an image for a state or an operation:

- The image must be accessible from the Configuration Manager workstation.
- The image height and width cannot exceed 16 pixels.

To define an image for a state or an operation:

1. In the Configuration Manager, access the State Configuration window for the family whose State Configuration contains a state for which you want to define an image.

2. If the **Enable State Functionality** check box is cleared, select it.

3. To define an image for a state, click the **States** tab.

   -or-

   To define an image for an operation, click the **Operations** tab.

4. Click the **Change Icon** link.

   The **Select an Image** window appears.

5. Navigate to the location containing the image file that you want to use, select the file, and click the **Open** button.

   The **State Configuration** window returns to focus, and the selected image appears next to the **Change Icon** link and in the **Icon** column in the grid.

6. Click the **Save** button.

   The image is saved to the database and will now appear in the Meridium APM Framework application in the appropriate places.

Deleting States

You *cannot* delete a state if it is the predecessor or successor state in an operation. Before you can delete a state this is used in an operation, you must delete the operation.

In addition, you can delete a state only if no records are currently set to that state.

To delete a state:

1. In the Configuration Manager, access the **State Configuration** window for the family containing the State Configuration whose state you want to delete.

2. Click the **States** tab.

3. In the grid on the **States** tab, select the row containing the state that you want to remove.

4. Click the **Delete State** link.

   A message appears, asking if you are sure that you want to delete the state.

5. Click the **Yes** button.

   The state is deleted.

6. Click the **Save** button.

   The state will no longer appear in the Meridium APM Framework application.

Deleting Operations

To delete an operation:

1. In the Configuration Manager, <u>access the **State Configuration** window</u> for the family containing the State Configuration whose operation you want to delete.

2. Click the **Operations** tab.

3. In the grid on the **Operations** tab, select the row containing the operation that you want to delete.

4. Click the **Delete Operation** link.

   A message appears, asking if you are sure that you want to delete the operation.

5. Click the **Yes** button.

   The operation is deleted.

6. Click the **Save** button.

   The operation will no longer appear in the Meridium APM Framework application.

Removing Images for States and Operations

To remove an image for a state or an operation:

1. In the Configuration Manager, [access the **State Configuration** window](#) for the family whose State Configuration contains a state or operation from which you want to remove an image.

2. To remove an image for a state, click the **States** tab.

   -or-

   To remove an image for an operation, click the **Operations** tab.

3. In the grid, select the row containing the state or operation from which you want to remove the image.

4. Click the **Clear Icon** link.

   The image is removed from the state and from the grid.

5. Click the **Save** button.

   The selected images will no longer appear in the Meridium APM Framework application.

Saving a State Configuration

Before a State Configuration can be saved successfully, it must include at least one state, and that state must be configured as the initial state.

To save a State Configuration:

- In the Configuration Manager, on the **State Configuration** window, after you have defined all desired states, operations, and rules, click the **Save** button.

  The State Configuration is saved.

About State Configuration Roles

A *State ConfigurationRole* is a group of Meridium APM Security Users who share similar responsibilities within a company and need to perform the same operations on records belonging to a given family.

For example, you might have several users who are responsible for reviewing and activating Asset Strategies, meaning that they need to change their states from Pending Review to Active using the *Make Active* operation. If so, you might create an *Asset Strategy Reviewers* State Configuration Role and assign all of these users to it. Then, assuming that you have configured State Configuration for the Asset Strategy family correctly (including assigning State Configuration Roles to the appropriate states), any of these users could activate any Asset Strategy record.

Together, the actions that you perform in the Configuration Manager and the APM Framework regarding State Configuration Roles dictate how state transitions will work for your users.

Within the Configuration Manager, you will need to:

- Create State Configuration Roles.
- **Assign State Configuration Roles to Security Groups**. For each State Configuration Role, you will need to assign one or more Security Groups to identify which users have permissions to assign and remove users to and from that State Configuration Role. For example, you might decide that only members of the *MI ASM Administrator* Security Group should have permissions to assign and remove users to and from the *Asset Strategy Reviewers* State Configuration Role. In this case, you would assign the MI ASM Administrator Security Group to the Asset Strategy Reviewers State Configuration Role.

  After you assign a Security Group to a State Configuration Role, via the Meridium APM Framework application, only members of that Security Group can assign and remove users to and from that State Configuration Role.

- **Assign State Configuration Roles to states**.
  - You can assign State Configuration Roles to *predecessor* states to define the group of users who are allowed to transition a record *from* that state.
  - You can assign State Configuration Roles to *successor* states to define *when* users who are allowed to transition a record to that state.

After an administrative user has defined the State Configuration for a family, via the Meridium APM Framework application, you can assign and remove Security Users to and from State Configuration Roles via the **Assign Users to a Role** window.

What Can I Do in the Configuration Manager?

The following table explains the actions that you can perform in the Configuration Manager regarding State Configuration Roles. Specifically, it lists the following actions, their results, and an example of each action and result:

- Assigning State Configuration Roles to *Security Groups*.
- Assigning State Configuration Roles to *predecessor states*.
- Assigning State Configuration Roles to *successor states*.

Assigning State Configuration Roles to Security Groups	
Action	**Example Action**
You assign a State Configuration Role to a Security Group.	You assign the State Configuration Role *Asset Strategy Reviewers* to the Security Group *MI ASM Administrators*.
Result	**Example Result**
Members of that Security Group can assign users to and from that State Configuration Role via the Meridium APM Framework application.	Members of the MI ASM Administrator Security Group can assign users to and from the Asset Strategy Reviewers State Configuration Role via the Meridium APM Framework application.
Assigning State Configuration Roles to Predecessor States Require a specific user to be assigned to a state check box = Selected	
Action	**Example Action**
You assign a State Configuration Role to a predecessor state. -and- You *select* the **Require a specific user to be assigned to a state** check box.	For the Asset Strategy family, you assign the State Configuration Role *Asset Strategy Reviewers* to the predecessor state *Pending Review*. -and- You *select* the **Require a specific user to be assigned to a state** check box.
Result	**Example Result**

Assigning State Configuration Roles to Security Groups	
A Security User can transition a record *out of* that state if he is assigned to: • That State Configuration Role. -and- • That state for that record.	Only the Security User who is assigned to the Pending Review state for a particular Asset Strategy record will be allowed to transition the Asset Strategy record *out of* the Pending Review state.

Assigning State Configuration Roles to Predecessor States Require a specific user to be assigned to a state check box = Cleared	
Action	**Example Action**
You assign a State Configuration Role to a *predecessor state*. -and- You *clear* the **Require a specific user to be assigned to a state** check box.	For the Asset Strategy family, you assign the State Configuration Role *Asset Strategy Reviewers* to the predecessor state *Pending Review*. -and- You *clear* the **Require a specific user to be assigned to a state** check box.
Result	**Example Result**
Any Security User who is assigned to that State Configuration Role can transition a record out of that state.	Any Security User who is assigned to the Asset Strategy Reviewers State Configuration Role can transition an Asset Strategy record *out of* the Pending Review state.

Assigning State Configuration Roles to Successor States Require a specific user to be assigned to a state check box = Selected	
Action	**Example Action**
You assign a State Configuration Role to a *successor state*. -and- You *select* the **Require a specific user to be assigned to a state** check box.	For the Asset Strategy family, you assign the State Configuration Role *Asset Strategy Reviewers* to the successor state *Active*. -and- You *select* the **Require a specific user to be assigned to a state** check box.
Result	**Example Result**

•

Assigning State Configuration Roles to Security Groups	
Any Security User can transition a record to that state, assuming that: • That user meets the criteria defined by the *predecessor* state selections. • The state is assigned to a [Security User who is assigned to that State Configuration Role]().	A Security User can transition an Asset Strategy record to the Active state if: • The user meets the criteria defined by the predecessor state. • The Active state is assigned to any Security User who is assigned to the Asset Strategy Reviewers State Configuration Role.

Assigning State Configuration Roles to Successor States	
Require a specific user to be assigned to a state check box = Cleared	
Action	**Example Action**
You assign a State Configuration Role to a *successor state*. -and- You *clear* the **Require a specific user to be assigned to a state** check box.	For the Asset Strategy family, you assign the State Configuration Role *Asset Strategy Reviewers* to the successor state *Active*. -and- You *clear* the **Require a specific user to be assigned to a state** check box.
Result	**Example Result**
Any Security User can transition a record to that state, assuming that the user meets the criteria defined by the *predecessor* state selections. In other words, because the check box is *cleared*, the State Configuration Role that is assigned to the successor state is ignored.	Any Security User that meets the criteria defined by the predecessor state selections can transition an Asset Strategy record to the Active state.

What Can I Do in the APM Framework?

The following table explains the actions that you can perform in the APM Framework regarding State Configuration Roles. Specifically, it lists the following actions, their results, and an example of each action and result:

- Assigning Security Users to State Configuration Roles.
- Assigning Security Users to states.

> Note: When you assign Security Users to states, you will not be interacting with State Configuration Roles in any way, but the outcome varies depending upon how State Configuration and State Configuration Roles have been configured for that family.

Assigning Security Users to State Configuration Roles	
Action	**Example Action**
You assign a Security User to a State Configuration Role.	You assign the Security User *John Smith* to the State Configuration Role *Asset Strategy Reviewers*.
Result	**Example Result**
That Security User can transition records: • *From* any predecessor states to which that State Configuration Role is assigned. • *To* any successor states that are assigned to a Security User that is assigned to that State Configuration Role.	John Smith can transition a record: • *From* any predecessor states to which the Asset Strategy Reviewers State Configuration Role is assigned. • *To* any successor states that are assigned to a Security User that is assigned to the Asset Strategy Reviewers State Configuration Role.
Assigning Security Users to States	
Action	**Example Action**
For a given record, you assign a Security User to a state.	For an Asset Strategy record, you assign the Security User *John Smith* to the *Draft* state.
Result	**Example Result**
Only that Security User can transition that record from that state.	*Only* John Smith can transition the Asset Strategy record *from* the Draft state.

Baseline State Configuration Roles

The following State Configuration Roles are available in the baseline Meridium APM database:

- MI ACA Owner
- MI ASI User
- MI ASM Analyst
- MI HA Facilitator
- MI HA Owner
- MI RBI Analyst
- MI RCM User
- MI Recommendation Management User
- MI SIS Administrator
- MI SIS Engineer

If you are upgrading from a previous version of Meridium APM in which *Security Groups* were eligible for state assignment, during the upgrade process, any Security Groups that were configured to be eligible for state assignment are used to create new State Configuration Roles. For example, if you previously configured the *MI ASM Administrator* Security Group to be eligible for assignment to a state in the Asset Strategy State Configuration, during the upgrade process, the *MI ASM Administrator* State Configuration Role will get created automatically.

Accessing the State Roles Window

To access the Roles window:

- In the Configuration Manager, on the main menu, click **Tools**, and then click **Manage State Roles**.

  The **State Roles** window appears.
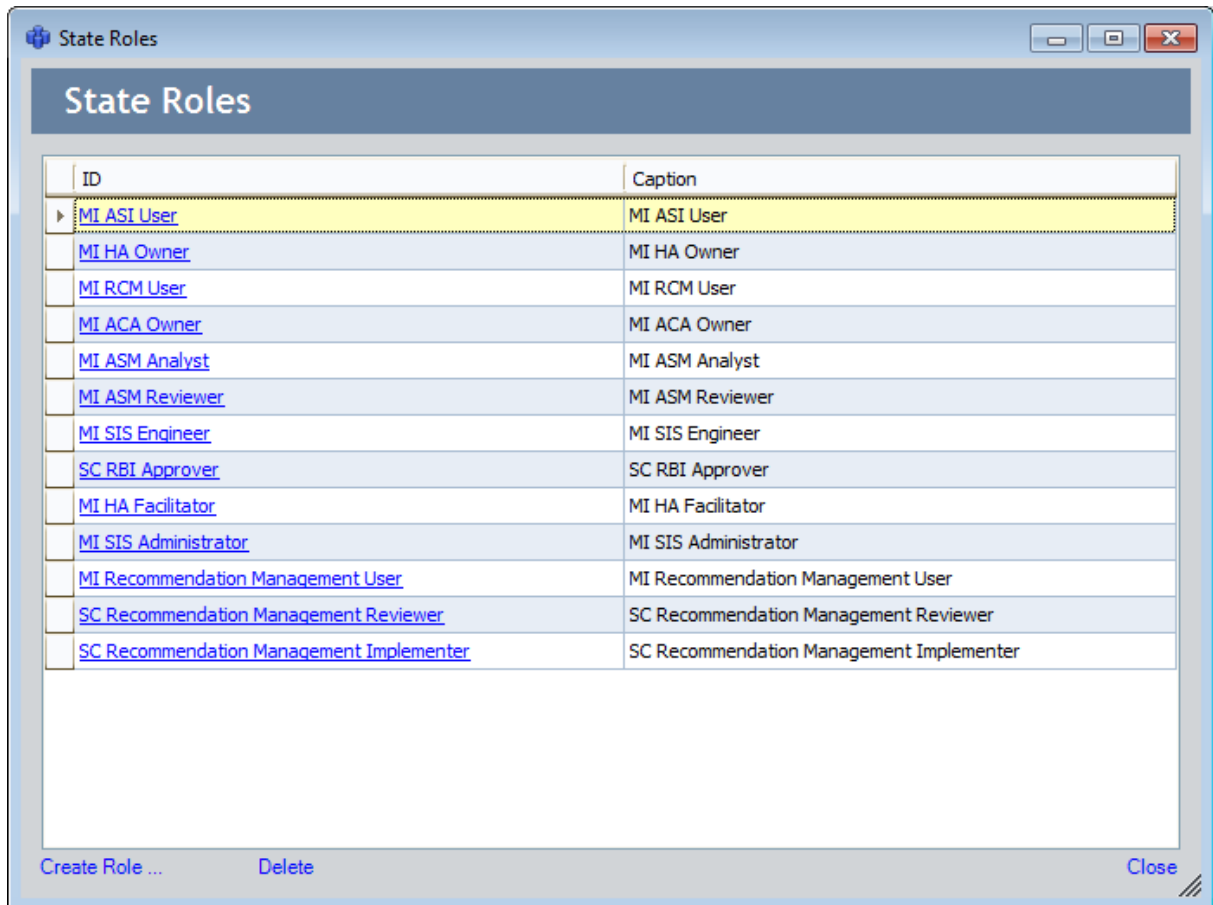
From the **State Roles** window, you can:

- [Create State Configuration Roles](#), which includes assigning those State Configuration Roles to Security Groups.
- [Modify State Configuration Roles](#).
- [Delete State Configuration Roles](#).

Creating State Configuration Roles

To create a new State Configuration Role:

1. In the Configuration Manager, access the **StateRoles** window.

   The **StateRoles** window appears.
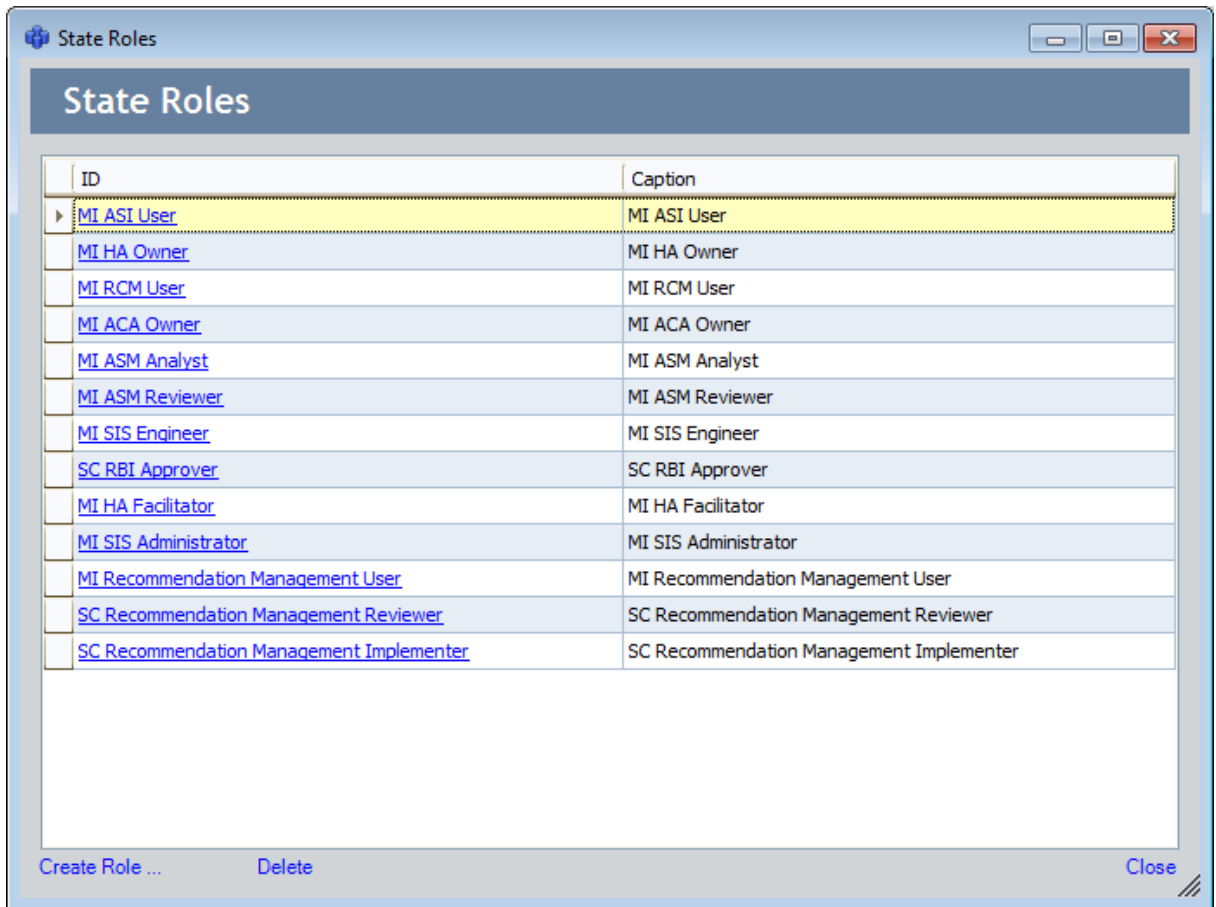
2. On the bottom left corner of the **StateRoles** window, click the **Create Role** link.

   The **Role** window appears.

•

3. In the **ID** text box, type an ID for the State Configuration Role.

4. In the **Default Caption** text box, type a caption for the State Configuration Role.

5. If desired, to the right of the **Default Caption** text box, click the [icon] button to manage translations for that string.

6. In the **Description** text box, type a description of the State Configuration Role.

> [i] **Note:** At this point, if you were to click the **Save** button, the State Configuration Role would be created successfully. These instructions assume, however, that as part of creating the State Configuration Role, you also want to assign Security Groups to the State Configuration Role to define the users who are allowed to assign and remove users to and from this State Configuration Role. The following instructions explain how to do so.

7. In the **Administrator Security Group(s)** section, click the **Modify** link.

   The **Select Groups** window appears.
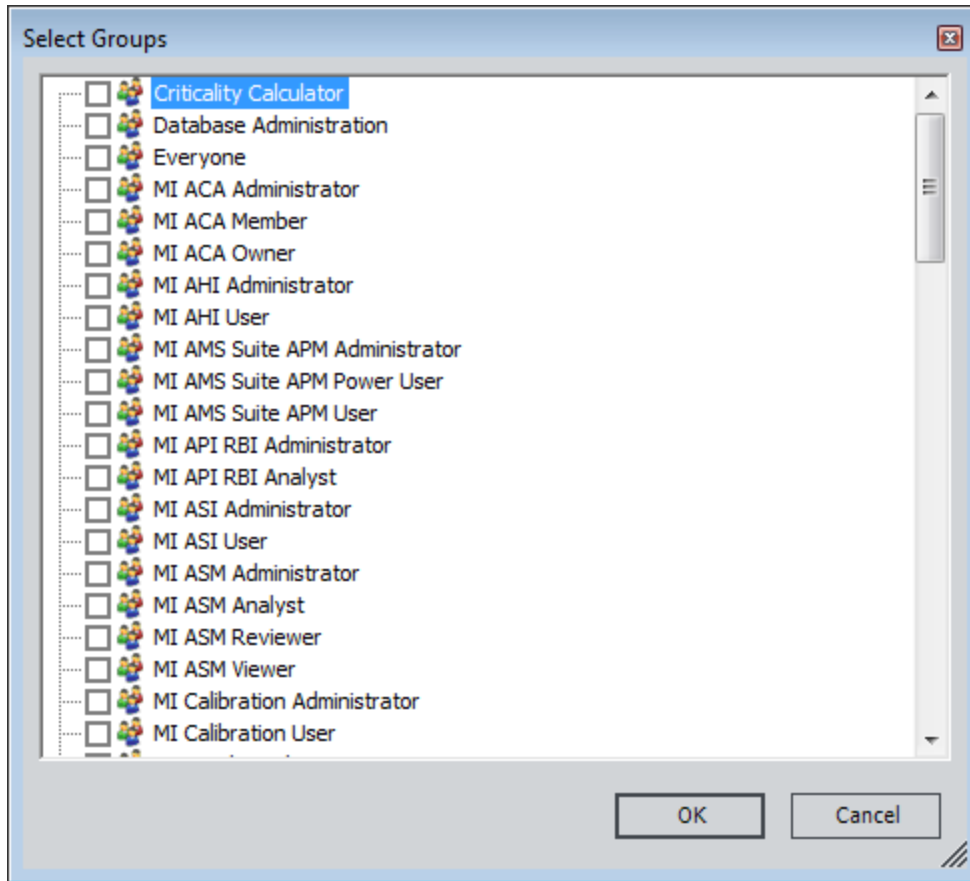
8. In the list, select the check box next to each Security Group whose members should be allowed to add and remove users to and from the State Configuration Role (via the APM Framework application).

9. Click **OK** to close the **Select Groups** window.

   The **Role** window returns to focus, displaying the selected Security Groups in the Administrator Security Group(s) section. In the following image, the *MI ACA Owner* Security Group appears in this section, indicating that members of this Security Group are allowed to assign and remove users to and from the MI ACA Owner State Configuration Role.
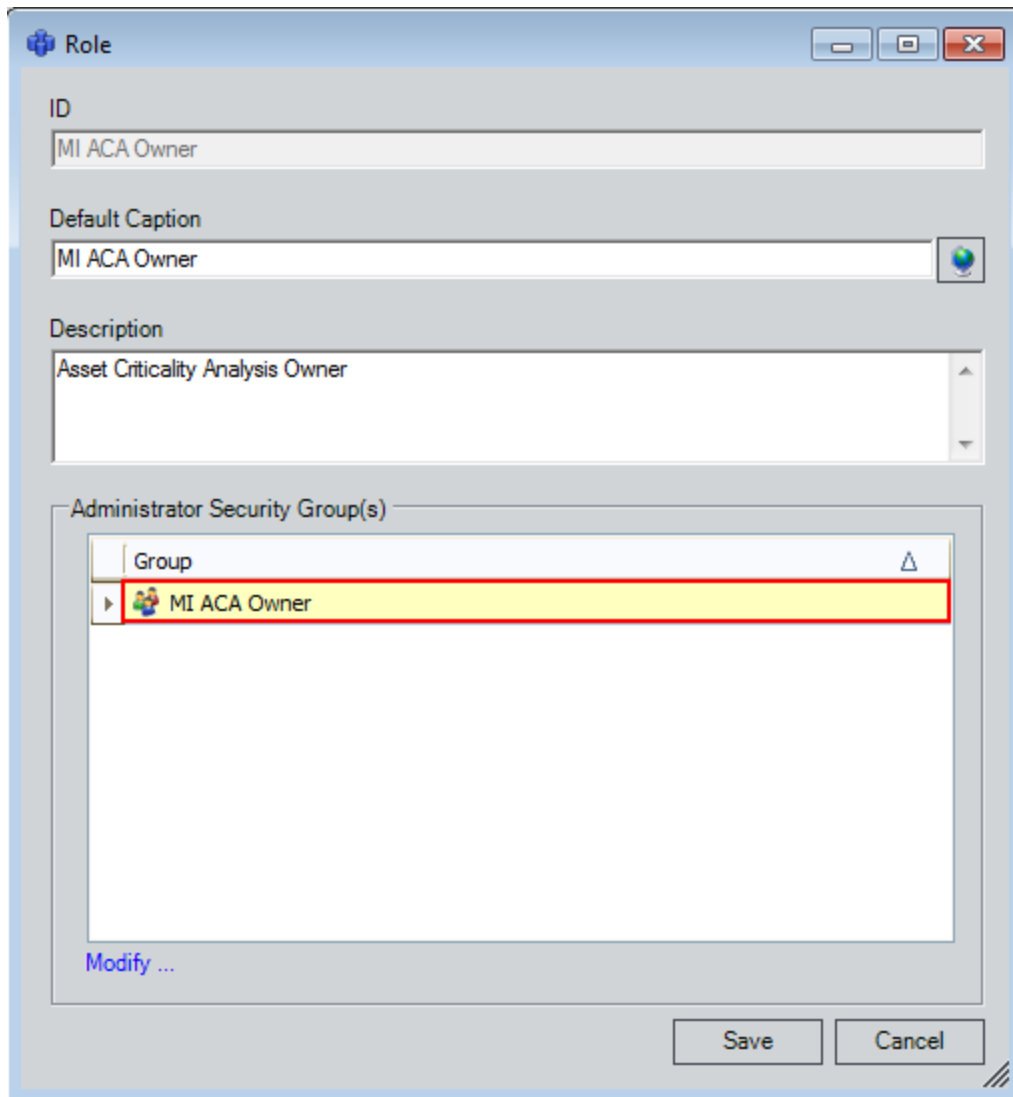
10. Click the **Save** button.

    The State Configuration Role is created, and the **StateRoles** window returns to focus.

11. Click the **Close** button to close the **StateRoles** window.

Modifying State Configuration Roles

You can modify the following aspects of a State Configuration Role:

- Default caption
- Description
- List of Security Groups whose members can assign and remove users to and from the State Configuration Role (via the APM Framework application).

You *cannot*, however, modify the ID of an existing State Configuration Role. If you want a State Configuration Role to have a different ID, you will need to delete the State Configuration Role and create a new one with the desired ID.

To modify a State Configuration Role:

1. In the Configuration Manager, access the **StateRoles** window.

   The **StateRoles** window appears.
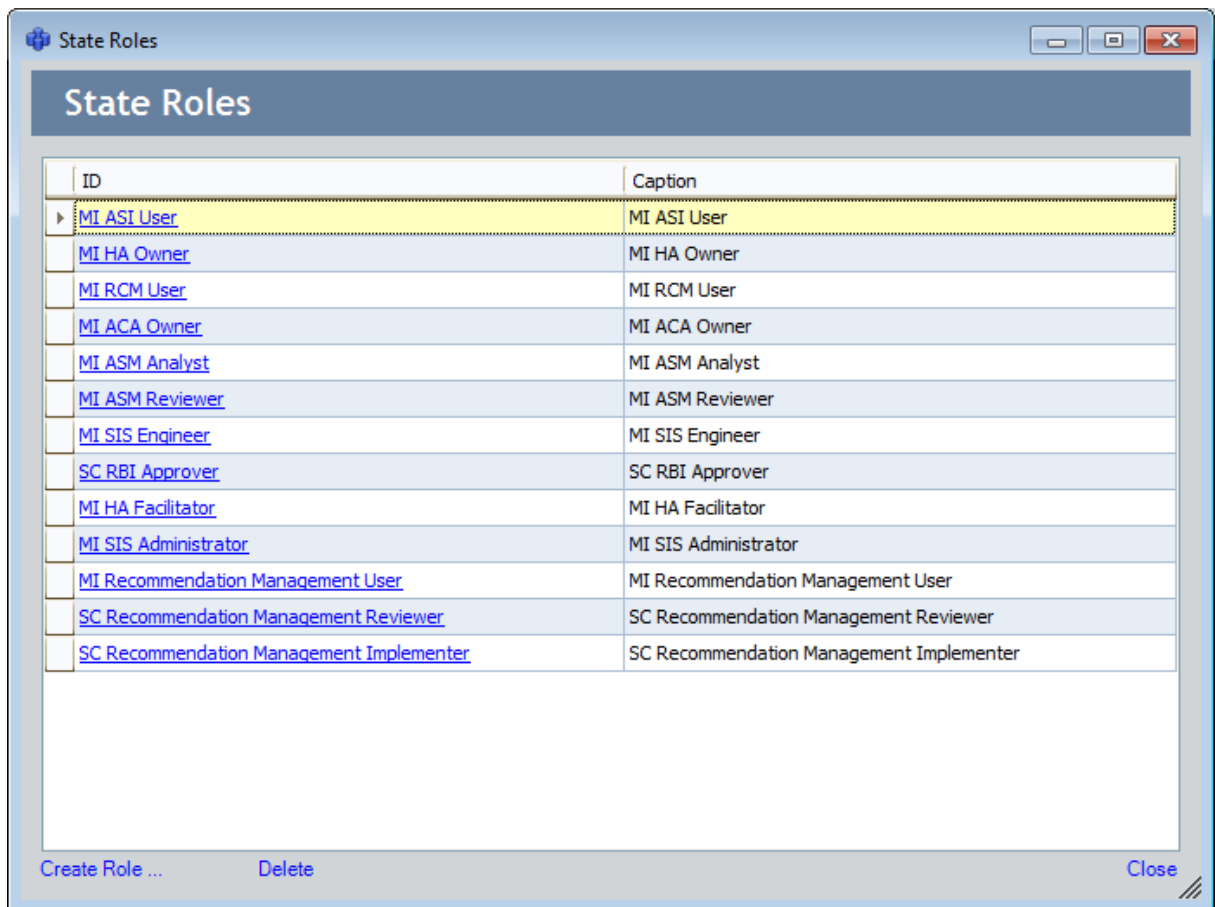
2. In the grid, in the row containing the State Configuration Role that you want to modify, click the hyperlinked ID.

   The **Role** window appears, displaying the selected State Configuration Role.
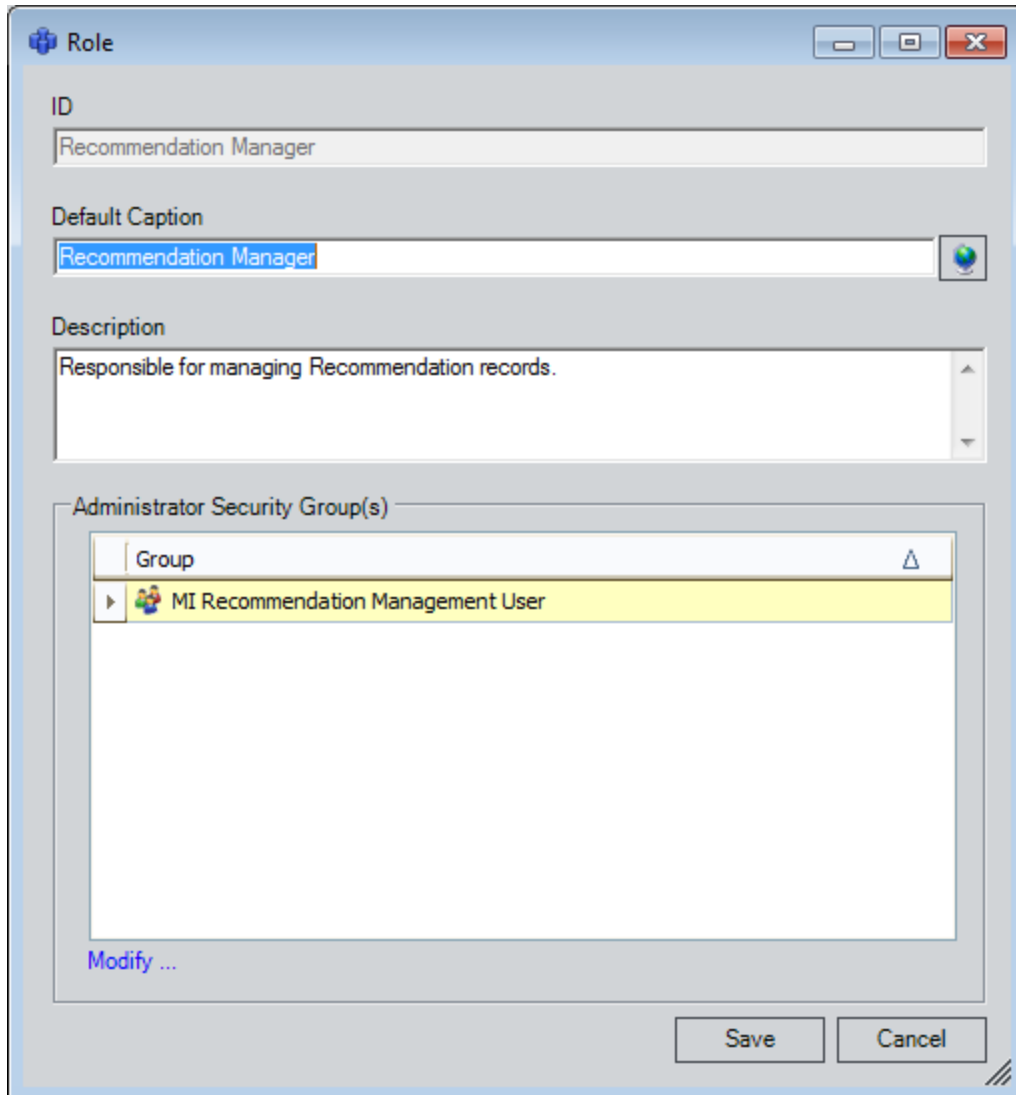
3. If you want to modify the default caption, in the **Default Caption** text box, modify the value.

4. If you want to modify the description, in the **Description** text box, modify the value.

5. If you want to modify the list of Security Groups whose members are allowed to assign and remove users to and from the State Configuration Role:

    a. In the **Administrator Security Group(s)** section, click the **Modify** link.

    The **Select Groups** window appears.
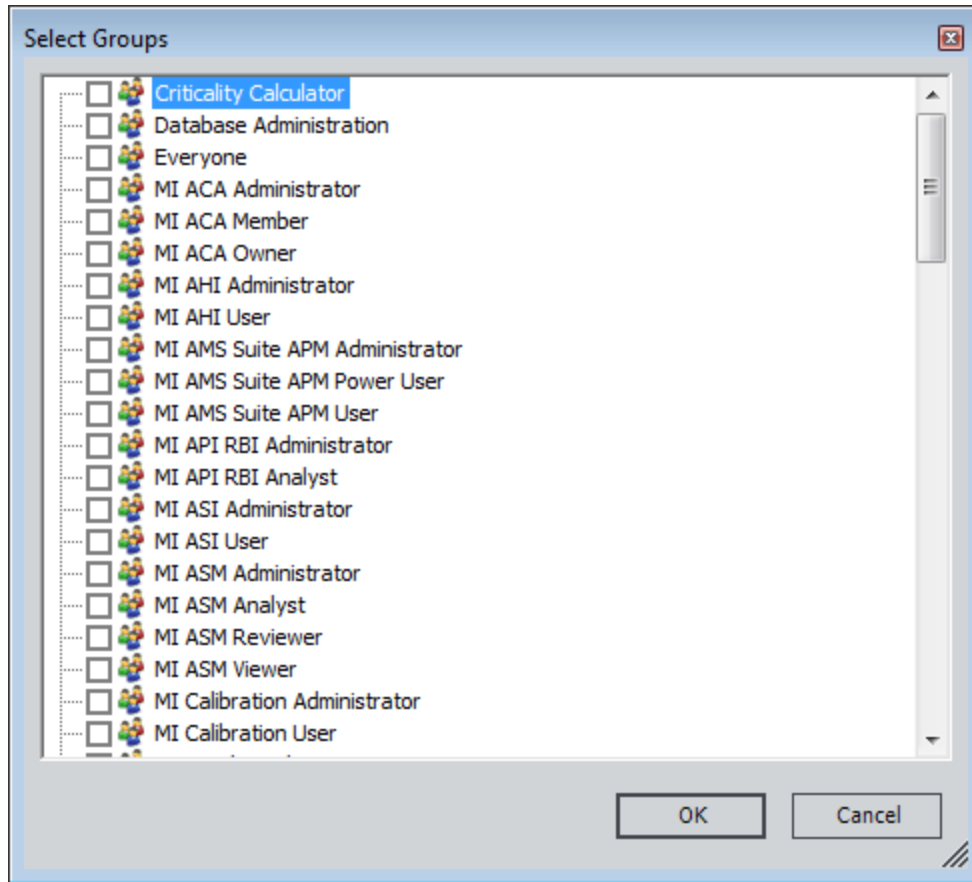
b. In the list, select the check box next to each Security Group whose members should be allowed to assign and remove users to and from the State Configuration Role (via the APM Framework application).

c. Click **OK** to close the **Select Groups** window.

6. On the **Role** window, click the **Save** button.

   The State Configuration Role is saved, and the **StateRoles** window returns to focus.

7. Click the **Close** link to close the **StateRoles** window.

Deleting State Configuration Roles

> ⓘ **Note:** You can delete a State Configuration Role only if it is not currently assigned to a state.

To delete a State Configuration Role:

1. In the Configuration Manager, access the **StateRoles** window.

   The **StateRoles** window appears.
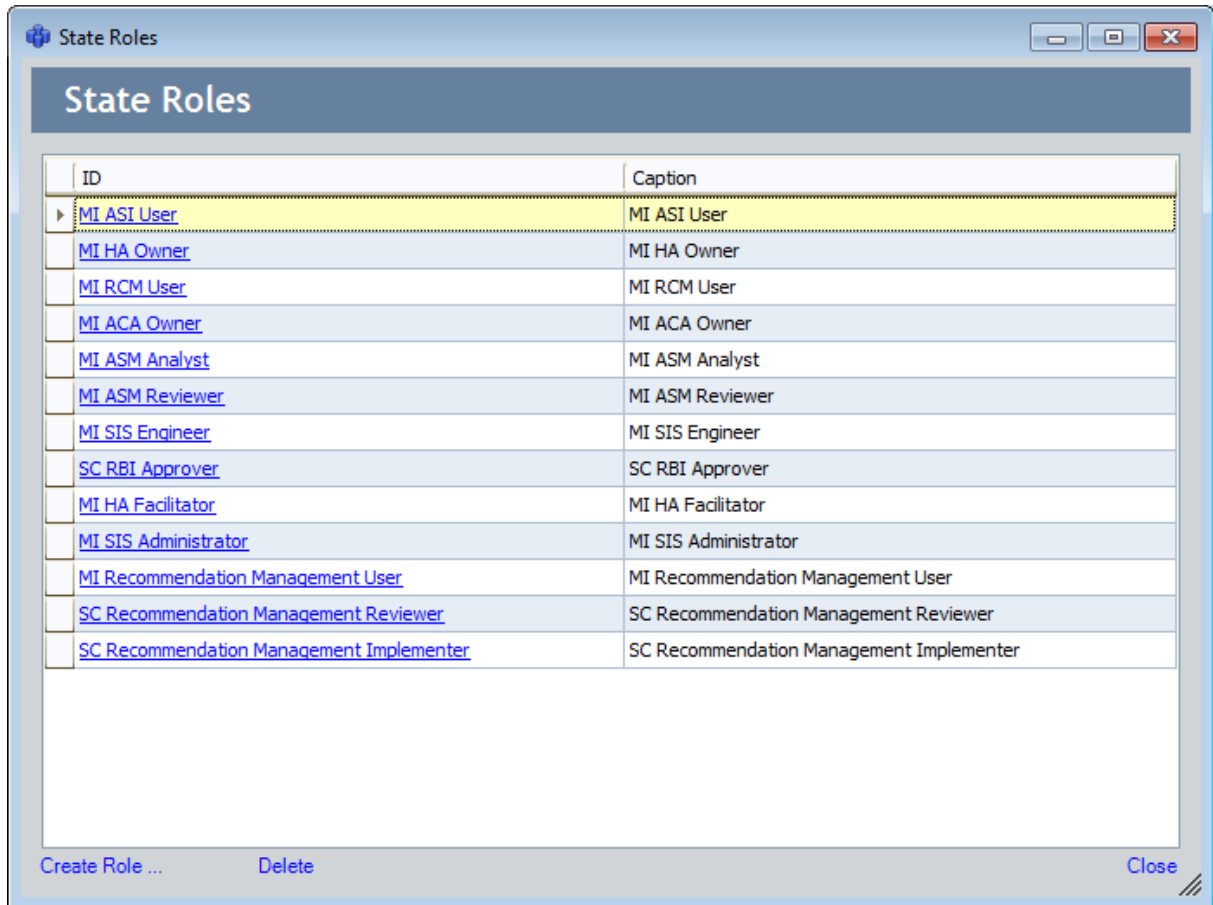
2. In the grid, select the row containing the State Configuration Role that you want to delete.

3. On the bottom left corner of the **StateRoles** window, click the **Delete** link.

   A message appears, asking if you are sure that you want to delete the State Configuration Role.

4. Click the **Yes** button.

   The State Configuration Role is deleted and removed from the grid.

•

Assigning State Configuration Roles to Security Groups

For each State Configuration Role that you create, you will need to assign one or more Security Groups to identify which users have permissions to assign and remove users to and from that State Configuration Role via the Meridium APM Framework application. For example, you might decide that only members of the *MI ASM Administrator* Security Group should have permissions to assign and remove users to and from the *Asset Strategy Reviewers* State Configuration Role. In this case, you would assign the MI ASM Administrator Security Group to the Asset Strategy Reviewers State Configuration Role.

After you assign a Security Group to a State Configuration Role, via the Meridium APM Framework application, only members of that Security Group can assign and remove users to and from that State Configuration Role.

You can assign State Configuration Roles to Security Groups:

- While you are creating the State Configuration Role.
- While you are modifying an existing State Configuration Role.

About Assigning State Configuration Roles to Predecessor States

You can assign State Configuration Roles to *predecessor* states to define the users who are allowed to transition a record *from* that state. For example, consider the State Configuration for the Meridium General Recommendation family, which contains the following predecessor and successor states for the Cancel operation:

- **Predecessor:** Reviewed
- **Successor:** Cancelled

You can confirm the predecessor and successor state for any operation by viewing the **Operations** tab of the **State Configuration** window when you access it for a family. In the following image, you can see on the **Operations** tab that for the *Cancel* operation, the predecessor state is *Reviewed* and the successor state is *Cancelled*.

In the following image, you can see on the **States** tab that the *Recommendation Manager* State Configuration Role is assigned to the predecessor state: *Reviewed*.
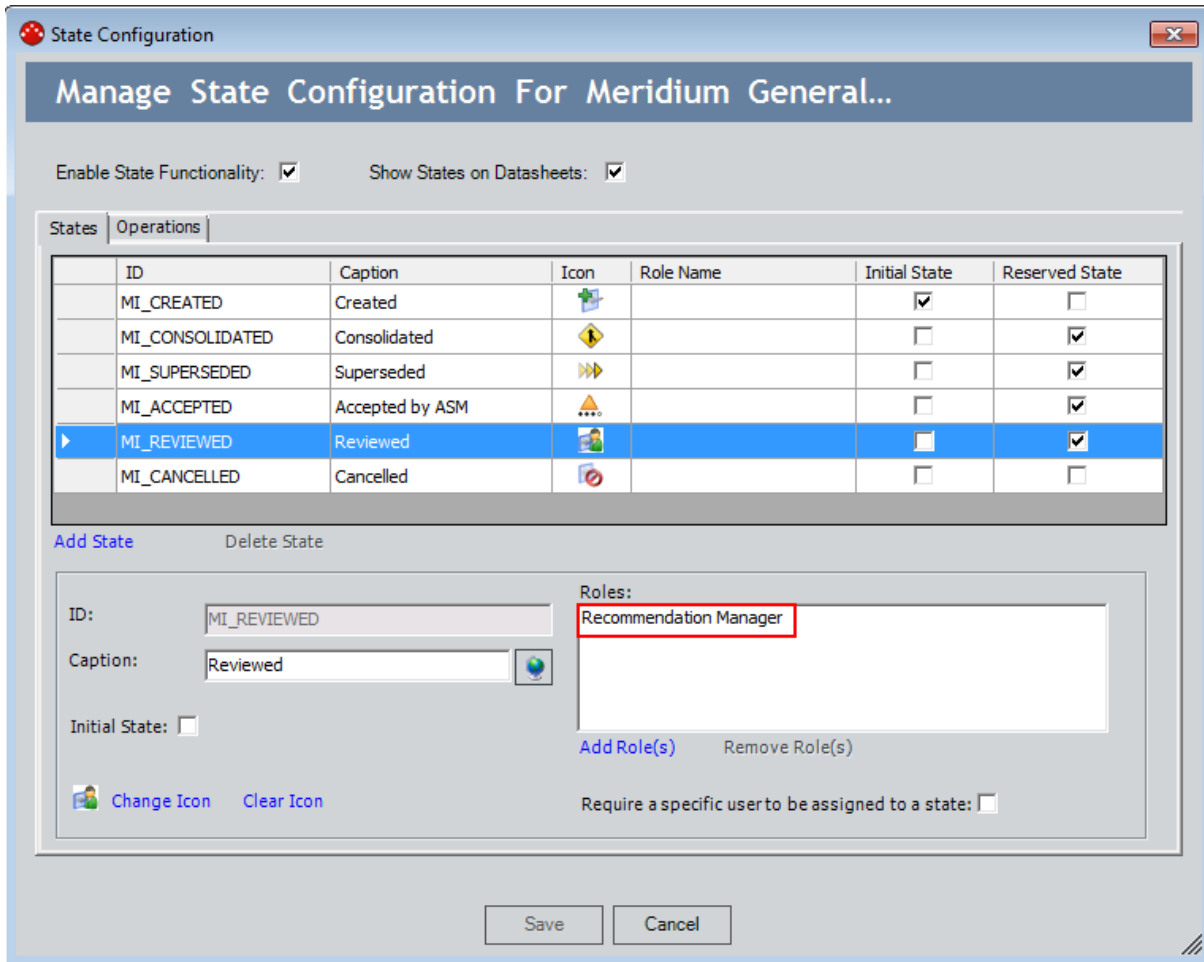
As a result of this assignment, any Security User who is assigned to the Recommendation Manager State Configuration Role will be allowed to transition a Meridium General Recommendation record *out of* the Reviewed state.

> ⓘ **Note:** Security Users can be assigned to State Configuration Roles via the Meridium APM Framework application.

Notice that in the preceding image where the Recommendation Manager Role is outlined, the **Require a specific user to be assigned to a state** check box is *cleared*. If it were selected, an additional layer of protection would be applied to Meridium General Recommendation records in the Reviewed state.

When the check box is *cleared*:

- *Any* Security User who is assigned to the Recommendation Manager Role can transition a record out of the Reviewed state.

When the check box is *selected*:

- For each individual record, only the Security User who is *assigned to the Reviewed state* for that record will be allowed to transition it out of the Reviewed state.

In other words, the State Configuration Role that is assigned to the state determines which Security Users are *eligible* to transition the record out of that state. Only the *specific Security User* that is assigned to the Reviewed state for a given record, however, will be allowed to move the record out of the Reviewed state.

> ⓘ **Note:** Security Users can be assigned to states via the Meridium APM Framework application.

About Assigning State Configuration Roles to Successor States

You can assign State Configuration Roles to *successor* states to define *when* users are allowed to transition a record to that state. For example, consider the State Configuration for the Meridium General Recommendation family, which contains the following predecessor and successor states for the Cancel operation:

- **Predecessor:** Reviewed
- **Successor:** Cancelled

You can confirm the predecessor and successor state for any operation by viewing the **Operations** tab of the **State Configuration** window when you access it for a family. In the following image, you can see on the **Operations** tab that for the *Cancel* operation, the predecessor state is *Reviewed* and the successor state is *Cancelled*.

In the following image, you can see on the **States** tab that the *Recommendation User* State Configuration Role is assigned to the successor state: *Cancelled*.
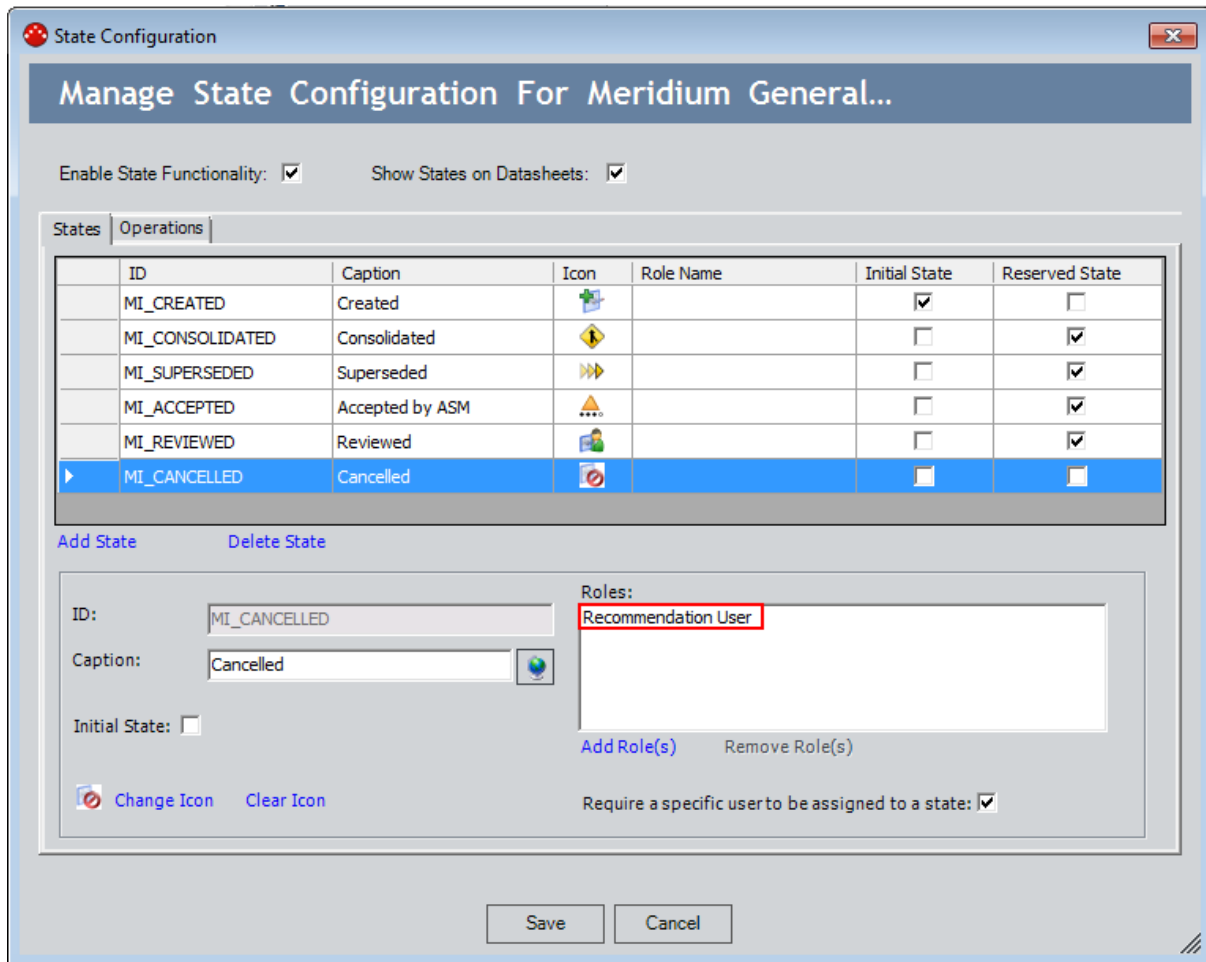
As a result of this assignment, a Security User can transition a Meridium General Recommendation record to the Cancelled state if the Cancelled state is assigned to a Security User who is assigned to the Recommendation User State Configuration Role.

> **Note:** Security Users can be assigned to State Configuration Roles via the Meridium APM Framework application.

Notice that in the preceding image where the Recommendation Manager Role is outlined, the **Require a specific user to be assigned to a state** check box is *selected*. If it were cleared, a layer of protection would be removed from Meridium General Recommendation records entering the Cancelled state.

When the check box is *selected*:

- A Meridium General Recommendation record can be set to the Cancelled state when any Security User who is assigned to the Recommendation User State Configuration Role is assigned to the Cancelled state.

When the check box is *cleared*:

- *No one* is required to be assigned to the Cancelled state in order for a user to transition it to the Cancelled state.

> 🗋 **Note:** Security Users can be assigned to states via the Meridium APM Framework application.

Examples of Assigning State Configuration Roles to States

Consider the following examples of [assigning State Configuration Roles to states](#) and selecting or clearing the **Require a specific user to be assigned to a state** check box. All examples assume that:

- You are working with the Meridium General Recommendation family.
- The predecessor state is *Reviewed*.
- The successor state is *Cancelled*.

In each example, the criteria in the table and the corresponding results in the bulleted list are color-coded so that you can see the connection between the settings and the results.

Example 1

State	Assigned Role	Require a specific user to be assigned to a state?
Reviewed	Recommendation Manager	Yes (check box is selected)
Cancelled	Recommendation User	Yes (check box is selected)

In this scenario, if a Meridium General Recommendation record is in the Reviewed state, you can change it to the Cancelled state if:

- You are assigned to the Recommendation Manager State Configuration Role.
- You are assigned to the Reviewed state.
- Any member of the Recommendation User Role is assigned to the Cancelled state.

Example 2

State	Assigned Role	Require a specific user to be assigned to a state?
Reviewed	Recommendation Manager	No (check box is cleared)
Cancelled	Recommendation User	No (check box is cleared)

In this scenario, if a Meridium General Recommendation record is in the Reviewed state, you can change it to the Cancelled state if:

- You are assigned to the Recommendation Manager State Configuration Role.

Note that:

•

- You are not required to be assigned to the Reviewed state.
- No one is required to be assigned to the Cancelled state.

Example 3

State	Assigned Role	Require a specific user to be assigned to a state?
Reviewed	Recommendation Manager	No (check box is cleared)
Cancelled	Recommendation User	Yes (check box is selected)

In this scenario, if a Meridium General Recommendation record is in the Reviewed state, you can change it to the Cancelled state if...

- You are assigned to the Recommendation Manager State Configuration Role.
- Any member of the Recommendation User State ConfigurationRole is assigned to the Cancelled state.

Note that:

- You are not required to be assigned to the Reviewed state.

Example 4

State	Assigned Role	Require a specific user to be assigned to a state?
Reviewed	Recommendation Manager	Yes (check box is selected)
Cancelled	Recommendation User	No (check box is cleared)

In this scenario, if a Meridium General Recommendation record is in the Reviewed state, you can change it to the Cancelled state if...

- You are assigned to the Recommendation Manager State Configuration Role.
- You are assigned to the Reviewed state.

Note that:

- No one is required to be assigned to the Cancelled state.

Example 5

State	Assigned Role	Require a specific user to be assigned **to a state?**
Reviewed	None	Yes (check box is selected)
Cancelled	None	Yes (check box is selected)

In this scenario, if a Meridium General Recommendation record is in the Reviewed state, *no one* will be able to change it to the Cancelled state because a specific user must be assigned to the Cancelled state but there are no State Configuration Roles assigned to this state, meaning that no one is even *eligible* to be assigned to the Cancelled state.

Example 6

State	Assigned Role	Require a specific user to be assigned **to a state?**
Reviewed	None	No (check box is cleared)
Cancelled	None	No (check box is cleared)

In this scenario, if a Meridium General Recommendation record is in the Reviewed state, *anyone* will be able to change it to the Cancelled state because there are no State Configuration Role restrictions and a specific user is not required to be assigned.

Assigning State Configuration Roles to States

To assign a State Configuration Role to a state:

1. In the Configuration Manager, [access the **State Configuration** window](#) for the family whose State Configuration contains a state to which you want to assign a State Configuration Role.

2. If the **Enable State Functionality** check box is cleared, select it.

3. Click the **States** tab.

4. In the grid, select the row containing the state to which you want to assign a State Configuration Role.

5. In the **Roles** area, click the **Add Role(s)** link.

   The **Add Roles** window appears.

6. Select the rows containing the State Configuration Roles that you want to assign to the state, and click **Add.**

   The selected State Configuration Roles appear in the **Roles** area.

7. Click the **Save** button.

   The selected State Configuration Roles are assigned to the state.

Accessing the Assign Users to a Role Window

To access the Assign Users to a Role window:

- In the Meridium APM Framework, on the main menu, on the **Tools** menu, click **StateRole Assignment**.

  The **Assign Users to a Role** window appears.
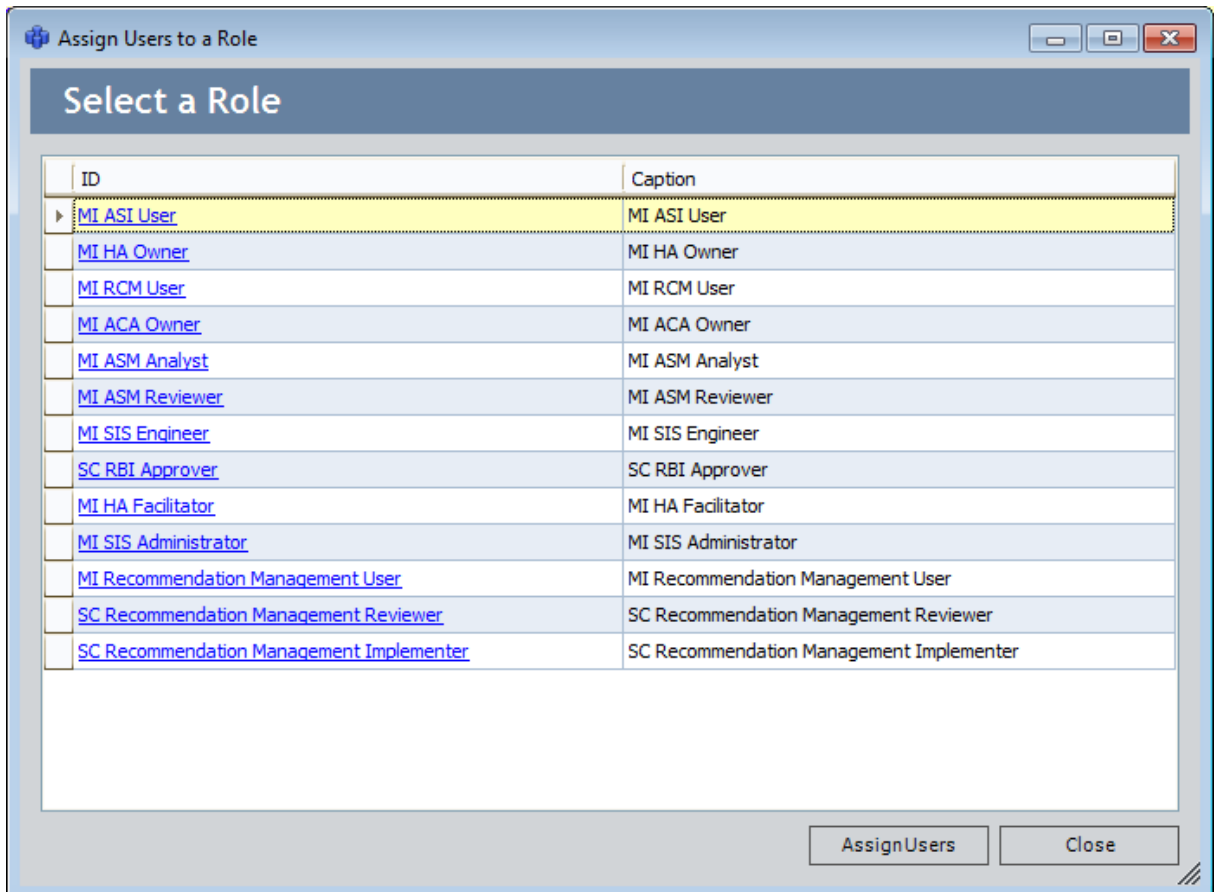
From the **Assign Users to a Role** window, you can:

- Assign Security Users to a State Configuration Role.
- Remove a Security User from a State Configuration Role.

Assigning Security Users to a State Configuration Role

To assign Security Users to a State Configuration Role:

1. In the Meridium APM Framework, access the **Assign Users to a Role** window.

   The **Assign Users to a Role** window appears.
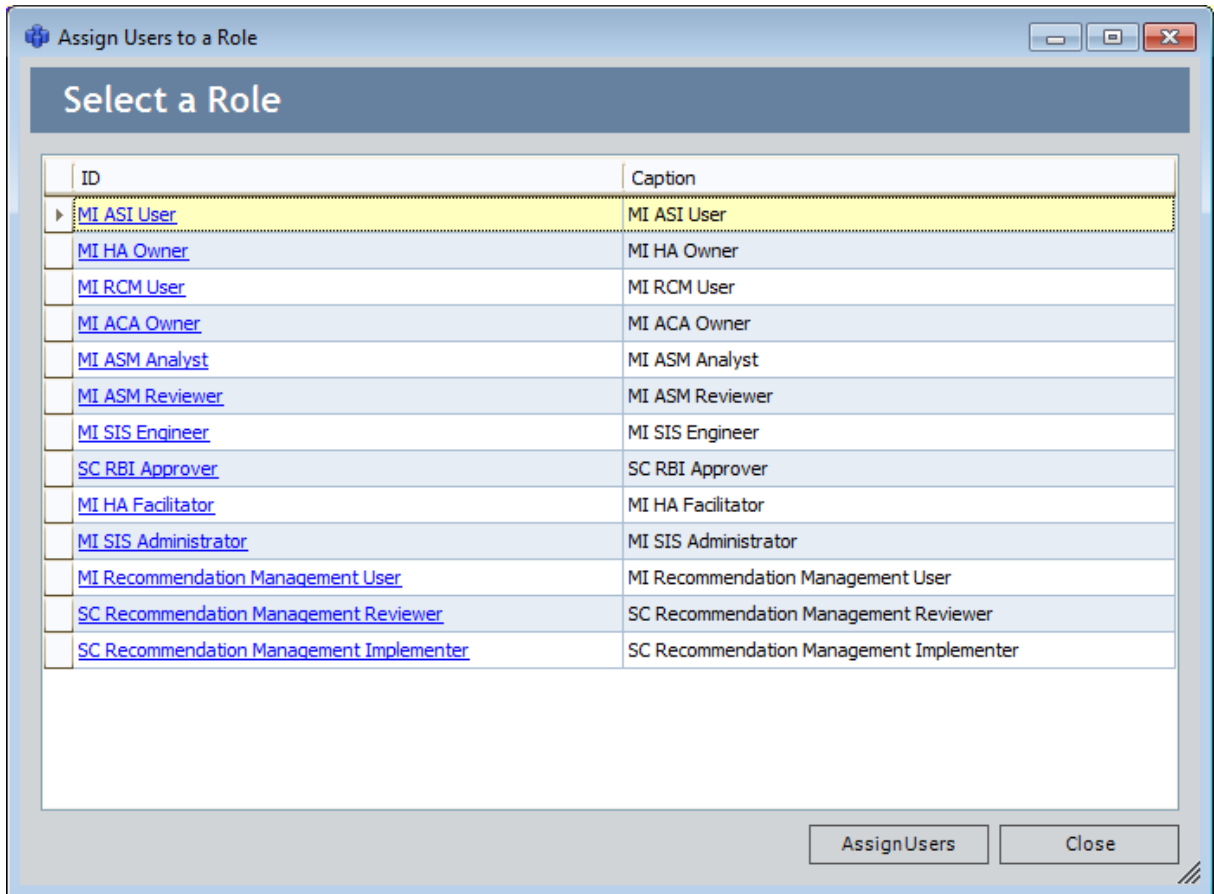
2. In the grid, in the row containing the State Configuration Role to which you want to assign Security Users, click the hyperlinked ID.

   > ⓘ **Note:** You could also click the **Assign Users** button while a row is selected.

   The **Assign Users** window appears.
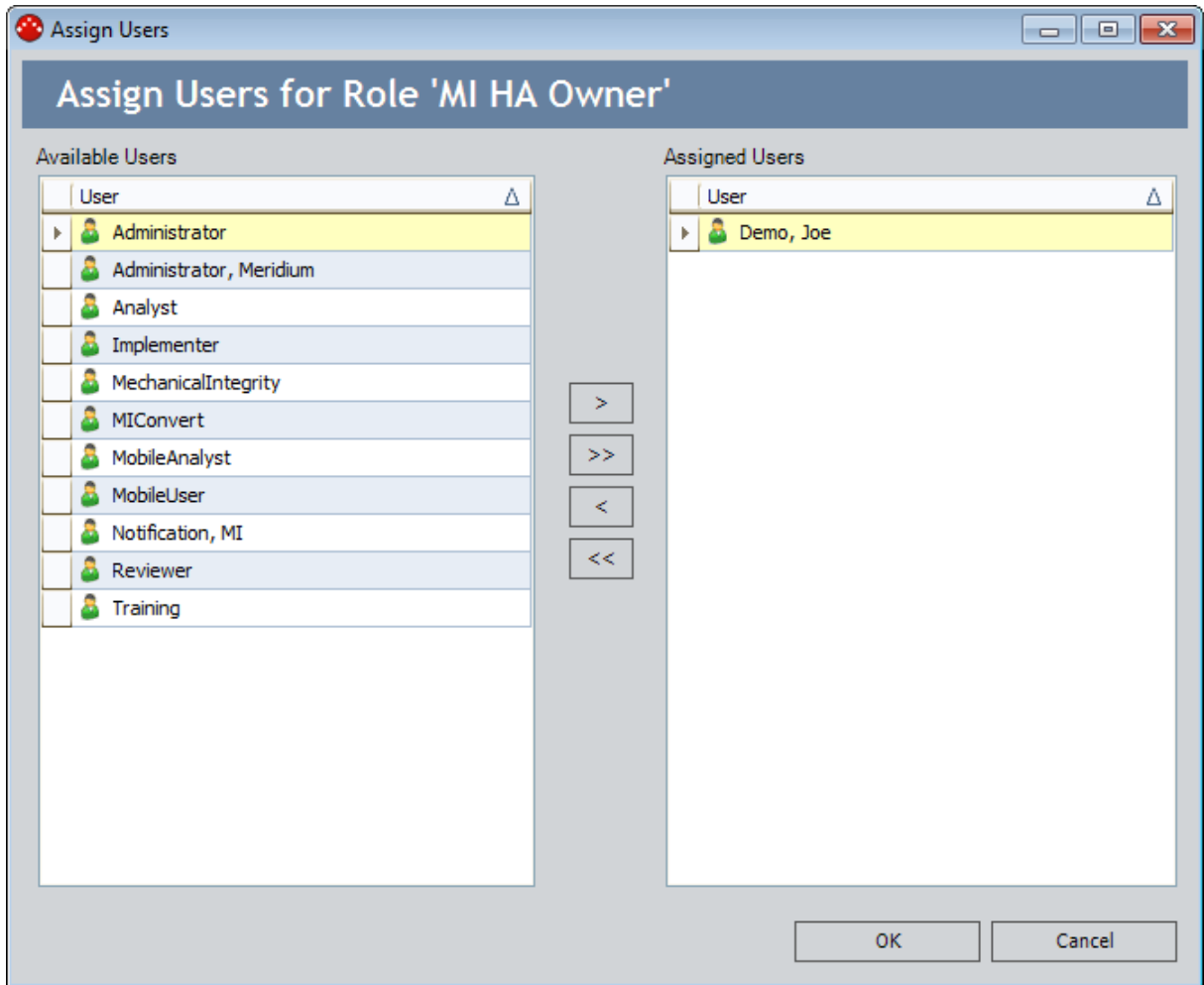
3. In the grid in the **Available Users** section, select the rows containing the Security Users that you want to assign to the State Configuration Role.

   > ☐ **Note:** If desired, you can assign ALL Security Users to the State Configuration Role by clicking the >> button.

4. Click the > button.

   The selected Security Users are added to the grid in the **Assigned Users** section.

5. Click **OK**.

   The **Assign Users to a Role** window returns to focus.

6. Click the **Close** button to close the **Assign Users to a Role** window.

Removing Security Users from a State Configuration Role

To remove Security Users from a State Configuration Role:

1. In the Meridium APM Framework, access the **Assign Users to a Role** window.

   The **Assign Users to a Role** window appears.
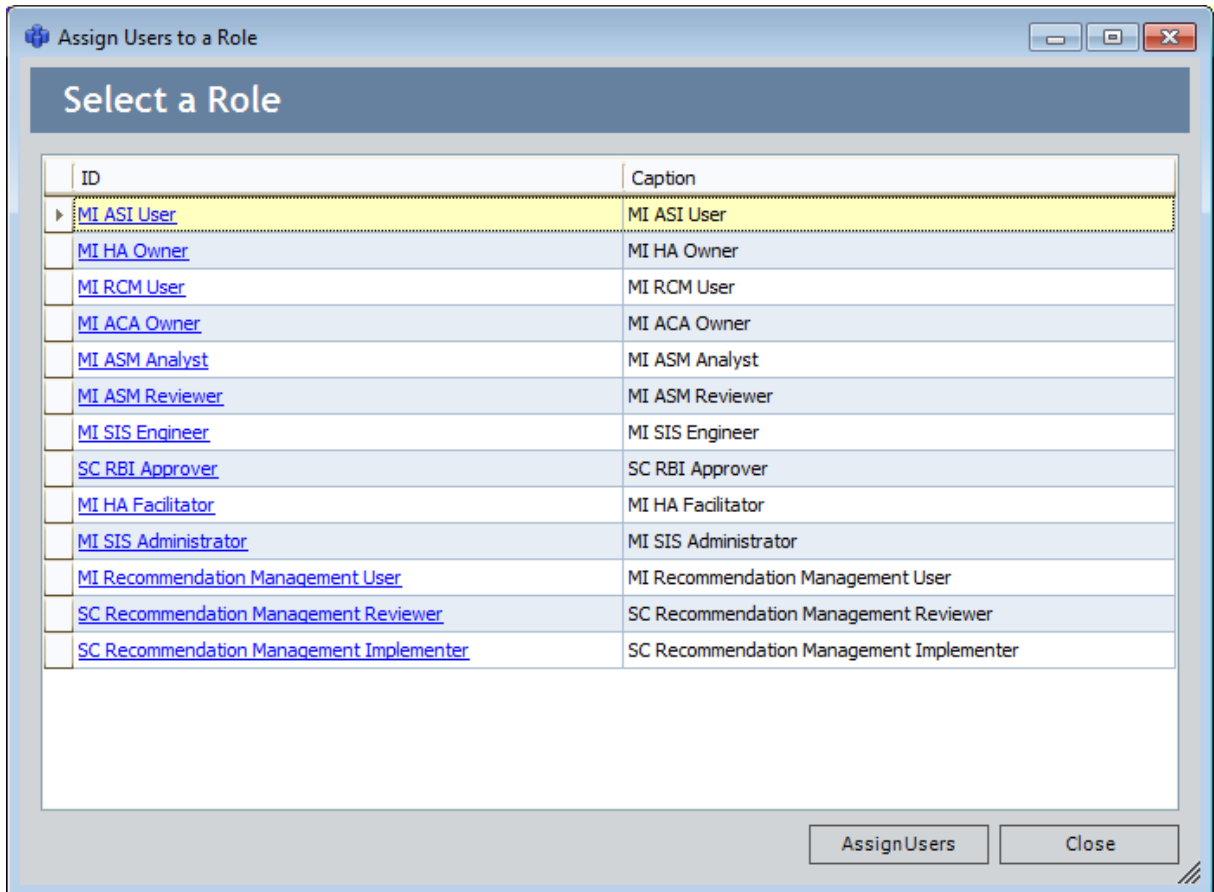
2. In the grid, in the row containing the State Configuration Role from which you want to remove Security Users, click the hyperlinked ID.

   > ⓘ **Note:** You could also click the **Assign Users** button while a row is selected.

   The **Assign Users** window appears.
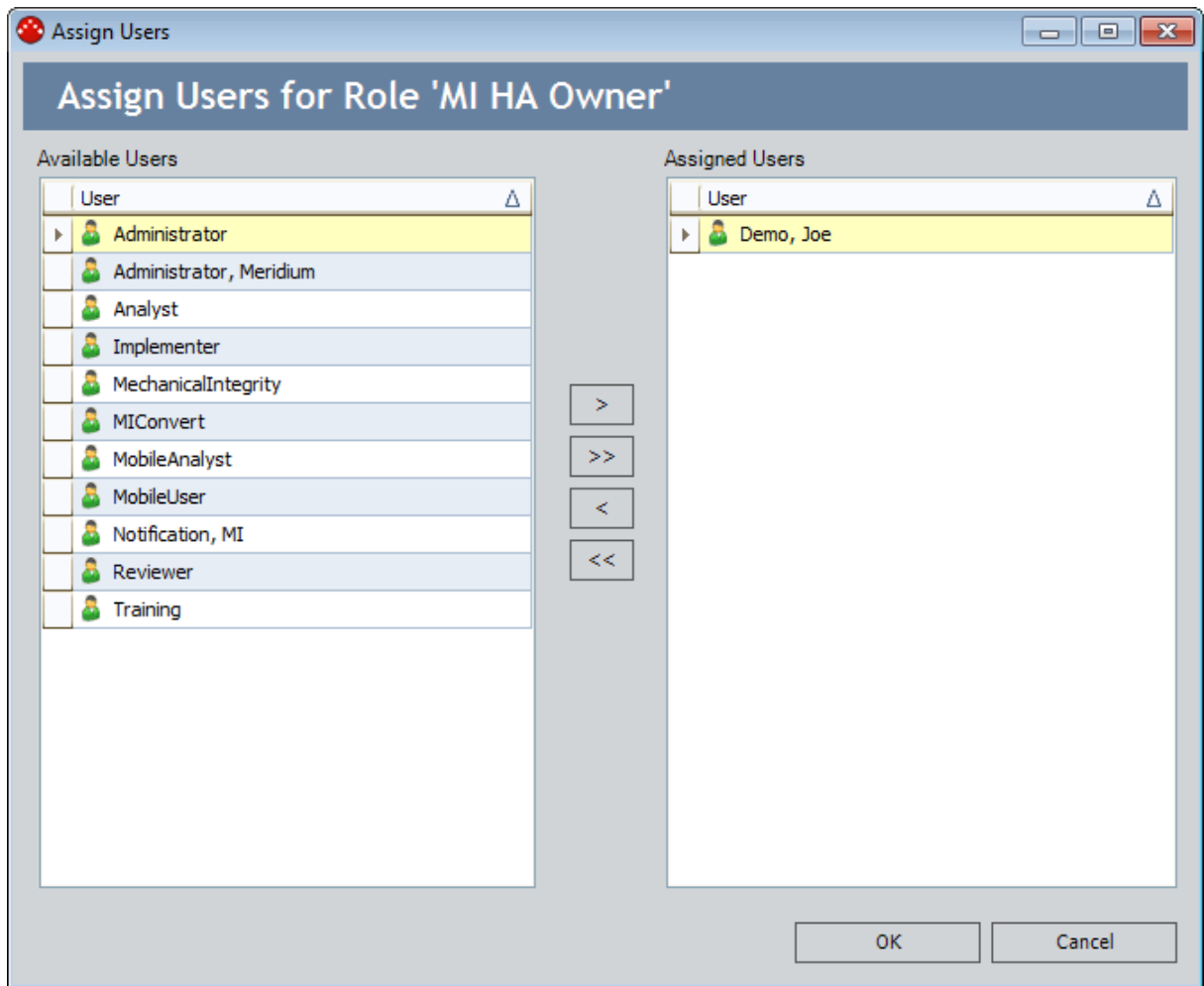
3. In the grid in the **Assigned Users** section, select the rows containing the Security Users that you want to remove from the State Configuration Role.

4. Click the [<] button.

> 🛈 **Note:** If desired, you can remove ALL Security Users from the State Configuration Role by clicking the [<<] button.

The selected Security Users are added to the grid in the **Available Users** section.

5. Click **OK**.

The **Assign Users to a Role** window returns to focus.

6. Click the **Close** button to close the **Assign Users to a Role** window.

About the Meridium APM Security Model

Security Users and Security Groups serve as the foundation for the Meridium APM security model.

- Security Users represent the individuals who will be accessing the Meridium APM system.
- Security Groups help organize individual Security Users according to their role within the Meridium APM system. You can use Security Groups to group together Security Users with similar roles and give all those users access to the same features.

After you have configured Security Users and Security Groups, you can configure the following types of security for them:

- Family-level security, where security privileges are defined at the entity family or relationship family level for individual Security Users and Security Groups.
- Data Filters, where privileges are assigned at the field level to determine which Security Users and Security Groups will be able to access individual entities (i.e., records), depending on the values that exist in the fields.

Meridium APM Security Data Model

The following image illustrates how families used by the Meridium APM security model are related to one another. In the following image, boxes represent entity families, and arrows represent relationship families that are configured in the baseline database.

•
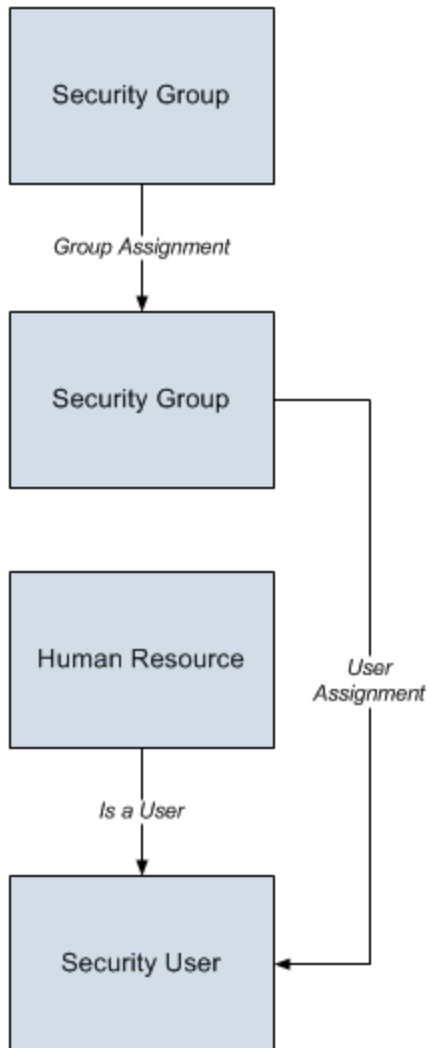
About Meridium APM Security Users

A Meridium APM Security User is an individual in your organization who has a user account for accessing the Meridium APM system. A Meridium APM Security User's account stores the user's ID and password, which is used for authenticating the Security User at login, and identifying information, such as the individual's name and contact information as well as details about that person's job within your organization.

Creating Security User accounts is the first step in configuring system security. After you have created Security Users, you can assign those Security Users to Security Groups, which provide you with a way of organizing Security Users according to their roles within the system. After you have set up Security Users and Groups, you can assign family-level privileges and set up Data Filters for those Security Users and Groups.

Each Security User account is actually a record that belongs to the Security User family. When you create a Security User in the Configuration Manager, three things happen:

- A Security User record is created.
- A corresponding Human Resource record is created.
- A link between the Security User record and the Human Resource record is created using the Is a User relationship family.

The following fields are defined in the baseline Security User family:

- User ID (SEUS_ID)
- LDAP User (SEUS_LDAP_USER)
- Password (SEUS_PASSWORD_TX)
- Status (SEUS_STAT_IND)
- Super User (SEUS_SUPER_USER_FLG)
- Query Privilege (MI_SEUS_QUERY_PRIV_CD)
- Culture (SEUS_CULTURE_ID)
- Conversion Set (SEUS_UOM_CONV_SET_CHR)
- Timezone (SEUS_TIME_ZONE_CHR)

The other fields that appear on the **Application User** dialog box are defined in the Human Resource family and are shared with the Security User family via a custom form. Because Security User records are actual records in the Meridium APM system, they can be searched, queried, viewed, and modified via the Meridium APM Framework application just like any other type of record. When you open a Security User record in the Meridium APM Framework application, you will see the fields that are defined in the Security User family itself and the fields that are defined in the Human Resource family. You can modify shared fields in either the Security User record or the Human Resource record, and your changes will be reflected in both records. Fields defined explicitly in the Security User family are not shared with the Human Resource family and can be modified via the Security User record only. Any changes that you make to a Security

User record in via the Meridium APM Framework application will be reflected in the Configuration Manager application as well.

> ⓘ **Note:** When you create a new Security User via the Configuration Manager application, the corresponding Human Resource record is created automatically. Alternatively, you can create Human Resource records in the Meridium APM Framework application or the Meridium APM Web Framework and promote them to Security Users via the Configuration Manager application. Security User records can also be created via the Meridium APM Framework application.

For the purposes of this documentation, we assume that each Security User record in your system will be associated with one individual in your organization. In this way, a Security User's account will store the identifying information for that person, such as the person's name, address, and job description. Note, however, that this does not imply a functional limitation to the number of users who can log in to the system using a single account. Any individual who knows the user name and password for a given Security User account can log in to the system as that Security User, even if another person is already logged in to the system with the same Security User account. Depending on the needs of your organization, you may opt to create Security User accounts that will be used by more than one individual.

Baseline Security Users

Meridium APM is shipped with the following baseline Security Users:

- **MIADMIN:** The baseline Meridium APM [Super User](). This user has access to all Meridium APM features. It is also used by the database upgrade process when a user is needed for performing an action, such as updating an entity. If desired, you can change the password for this user.

  > ⓘ **Note:** By default, the **MIADMIN** user is a member of the MI Catalog Administrator, MI RM Administrator, MI SM Administrator, Everyone, and MIAdmin Security Groups. If you modify the default membership, note that this user must *at least* be a member of the Everyone Security Group and *cannot* be a member of the MI Configuration Role or MI Security Role Security Group.

- **MIConvert:** This Security User is not currently used.

- **MINotification:** A Security User that is used for executing some scheduled items. This Security User has a blank password.*Do not add a password to this user account.*

You cannot delete these Security Users from the database. You should not deactivate these Security Users, as they are needed for the proper functioning of Meridium APM.

> ⓘ **Note:** Meridium APM is also shipped with a set of [baseline Security Groups]().

About Meridium APM Super Users

Any Meridium APM Security User can be designated a Meridium APM Super User, which is a user who can access all features in Meridium APM. Super Users are special because they do not require additional Security Group membership to gain access to the features in Meridium APM. Super users can access:

- The Configuration Manager.
- The Meridium APM Framework application.
- The Meridium APM Web Framework.
- The URL Manager.

> Note: Any user who installs the Meridium APM Administrative Applications will have full access to the Schedule Manager and the Data Source Manager.

In addition, they do not need family-level privileges in order to create, edit, and delete records. Rather, Super Users have automatic access to ALL records in the database.

You can create Super Users by selecting the **Super User** check box on the User Information tab of the **Application User** dialog box.

Viewing a List of Existing Security Users

To view a list of Security Users that already exist in the system:

- In the Configuration Manager, on the Configuration Manager main menu, click **Security**, and then click **Users**.

  The **Application Users** window appears, displaying the list of Security Users that are already defined in the database.
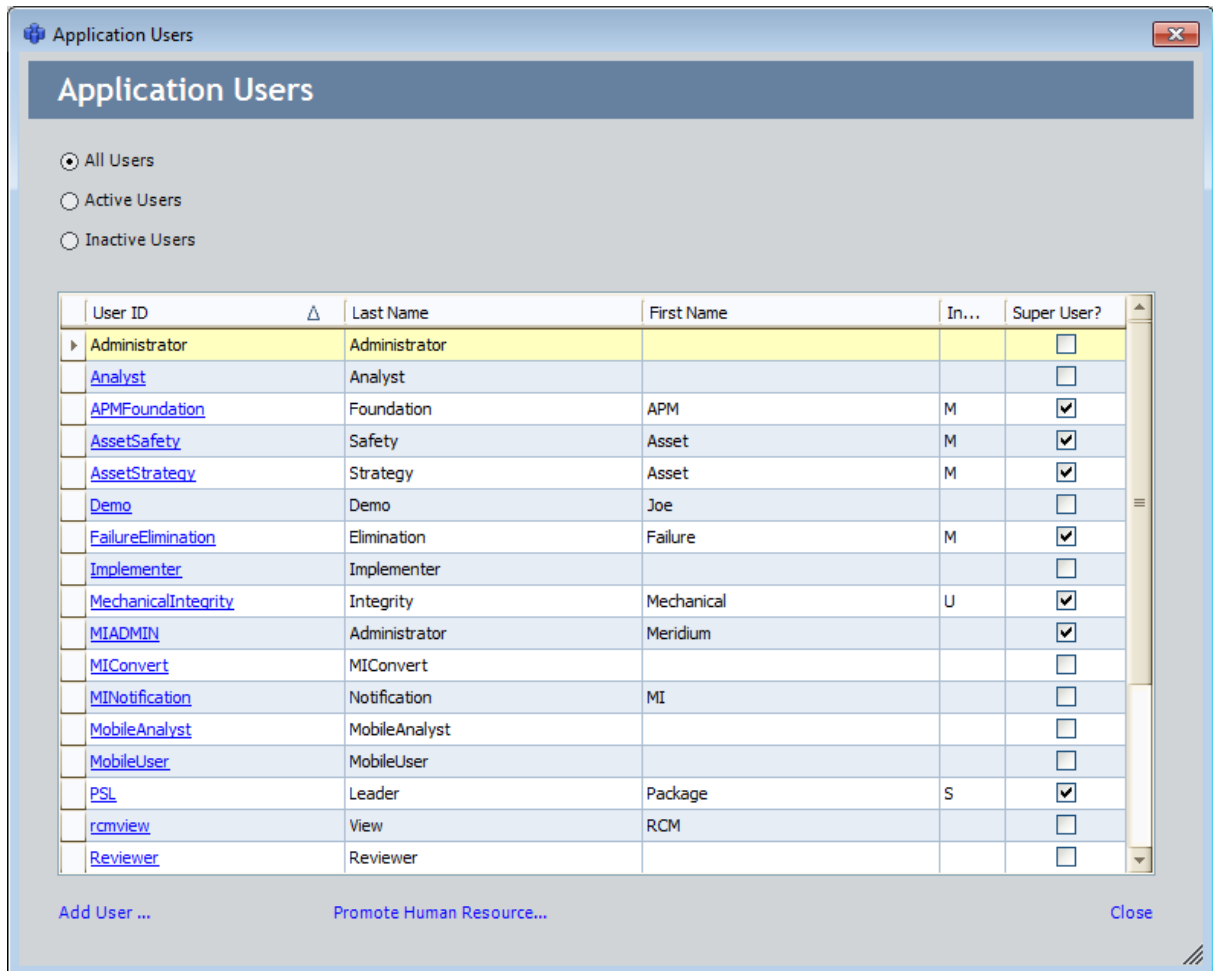
From the **Application Users** window, you can add a new Security User, edit an existing Security User, or promote a human resource to a Security User. Additionally, on the **Application Users** window, you can sort and filter the list based on various criteria. In addition to the standard grid customization options, which are available from a shortcut menu that appears when you click any column header, you can use the following filter options, which appear above the grid:

- **All Users:** Displays ALL Security Users in the grid, regardless of the value in the Status field.

- **Active Users:** Displays in the grid only the Security User records that contain the value *Active* in the Status field.

- **Inactive Users:** Displays in the grid only the Security Users records that contain the value *Inactive* in the Status field.

•

Creating a New Security User from Scratch

To create a new Security User from scratch:

1. In the Configuration Manager, on the **Application Users** window, click the **Add User** link.

   The **Application User** dialog box appears, displaying blank fields that you can use for defining the user properties.

2. Configure the desired Security User properties.

   > ⓘ **Note:** You must define at least a **User ID** and a **Last Name** for the user before you can save the user to the database.

3. When you are finished, click the **Save** button.

   The new Security User is saved to the database.

Creating a New Security User by Copying an Existing Security User

To create a new Security User by copying an existing Security User:

1. In the Configuration Manager, on the **Application Users** window, click the hyper-linked User ID of the Security User that you want to copy.

   The **Application User** dialog box appears, displaying the information that is currently associated with the selected Security User.

2. At the bottom of the window, click the **Copy User** link.

   A message appears, indicating that if you proceed with the copying process, a new Security User will be created and *saved* automatically, and you will not be able to cancel the operation.

3. Click the **Yes** button.

    A new Security User is created as a copy of the selected Security User, and the information for the new Security User appears on a new **Application User** window.

    All values from the source Security User are copied to the new Security User except for:

- First Name
- Middle Initial
- Email Address
- Password

    > **Note:** The password is set as a blank password.

    In addition:

- The User ID is copied and appended with _ *<n>*, where *<n>* is a number (starting with 0) representing the number of Security Users that have been created from the source Security User.
- The last name is set to *Last Name Copied*.
- The new Security User will belong to all Security Groups to which the source Security User belongs. The new Security User will not inherit family-level privileges that are granted specifically to the source Security User.

4. Modify the User ID as appropriate.

5. Specify the password as appropriate. If you do not modify the password, when you save the Security User, the Security User will be able to log in using only their User ID *without* a password.

6. Modify the last name as appropriate.

7. Modify any other values that are invalid for the new Security User.

8. Click the **Save** button.

    The new Security User is saved.

Modifying Security Users

To edit the properties of an existing Security User:

1. In the Configuration Manager, on the **Application Users** window, click the hyper-linked User ID of the Security User that you want to modify.

   The **Application User** dialog box appears, displaying the information that is currently associated with the selected Security User.

2. Modify the Security User properties as desired.

3. Click the **Save** button.

   Your changes are saved to the database. If you changed the password for the Security User, after your changes are saved, a message will appear, indicating that the password for the scheduled items associated with that user account have been updated with the new password.

About Human Resource and Security User Records

Security Users are stored in records that belong to the Security User family, which is related to the Human Resource family through the Is a User relationship family. When you create a new Security User record, either in the Configuration Manager application or in the Meridium APM Framework application:

- A corresponding Human Resource record is created automatically.
- The Human Resource record is linked to the Security User record through the Is a User relationship.

> ⓘ **Note:** You cannot remove the link between a Security User record and a Human Resource record. This means that you cannot delete a Human Resource record via the Meridium APM Framework application if the associated user is also a Security User in the Configuration Manager.

The following fields are defined in the baseline Security User family:

- User ID (SEUS_ID)
- Password (SEUS_PASSWORD_TX)
- LDAP User (SEUS_LDAP_USER)
- Super User (SEUS_SUPER_USER_FLG)
- Status (SEUS_STAT_IND)
- Culture (SEUS_CULTURE_ID)
- Conversion Set (SEUS_UOM_CONV_SET_CHR)
- Timezone (SEUS_TIME_ZONE_CHR)

The other fields that appear in Security User records are actually defined in the Human Resource family and are "shared" with the Security User family via a custom form. The form that is used in the Configuration Manager for recording data for a Security User is identical to the datasheet configured for the Security User family and similar to the default datasheet for the Human Resource family. When you search on the Human Resource or Security User family in the Meridium APM Framework application, you will see these datasheets in the Record Manager if the default configuration has not been modified.

Note that the fields defined in the Security User family do not appear in the Human Resource record. Fields defined in the Human Resource family, however, appear on the Security User form. You can modify shared fields in either the Security User record or the Human Resource record, and your changes will be reflected in both records. Fields defined explicitly in the Security User family can be modified via the Security User record only. Any changes that you make to a Security User record in via the Meridium APM Framework application will be reflected in the Configuration Manager application as well.

In some workflows where you work with analysis records, such as the RCA workflow, you will also work with team members, which are simply Human Resource records that are linked to the analysis record. Because each Security User record also has a corresponding Human Resource record, all Security Users are available to be linked to the analysis. In addition, you can create new Human Resource records and link them to the analysis.

Any Human Resource record that you create via the Meridium APM Framework application will not automatically have a corresponding Security User record. Via the Configuration Manager, however, you can promote the person associated with the Human Resource record to a Security User via the **Promote Resource** link on the **Application Users** dialog box. Alternatively, you can promote the person to a Security User via the **Promote Resource** button in the Human Resource record. Note, however, that this button appears only to Super Users. After clicking the **Promote Resource** link or button, you will be prompted to define the User ID, password, and Super User status for the Security User, which will create a Security User record for that user.

> ⓘ **Hint:** To keep track of Human Resource records that you have added that do not have a corresponding Security User in the Configuration Manager, you can create a query where you can search for all Human Resource records that are not linked to a Security User record.

Promoting Human Resources from the Configuration Manager

When you create a Security User via the Configuration Manager, the following occurs:

- A Security User record is created.

- A corresponding Human Resource record is created.

- The Human Resource record is linked to the Security User record through the **Is a User** relationship.

If you create a Human Resource record via the Meridium APM Framework application the Meridium APM Web Framework, that record is not automatically linked to a Security User record. This means that the person associated with the Human Resource record will not appear in the list of Security Users in the Configuration Manager. You can, however, *promote* that Human Resource record to a Security User. By doing so, the user will appear in the list of Security Users and a Security User record will be created for that person. By promoting a Human Resource record, you create a corresponding Security Record and establish a link between the two records via the **Is a User** relationship.

> ⓘ **Note:** You can also promote Human Resource records in the Meridium APM Framework application. For details, see the Meridium APM Framework Help.

To promote a Human Resource record to a Security User:

1. In the Configuration Manager, on the **Application Users** window, click the **Promote Resource** link.

   The **Import/Export Results Resources** dialog box appears, displaying a row representing each Human Resource record for which a related Security User record does not yet exist.
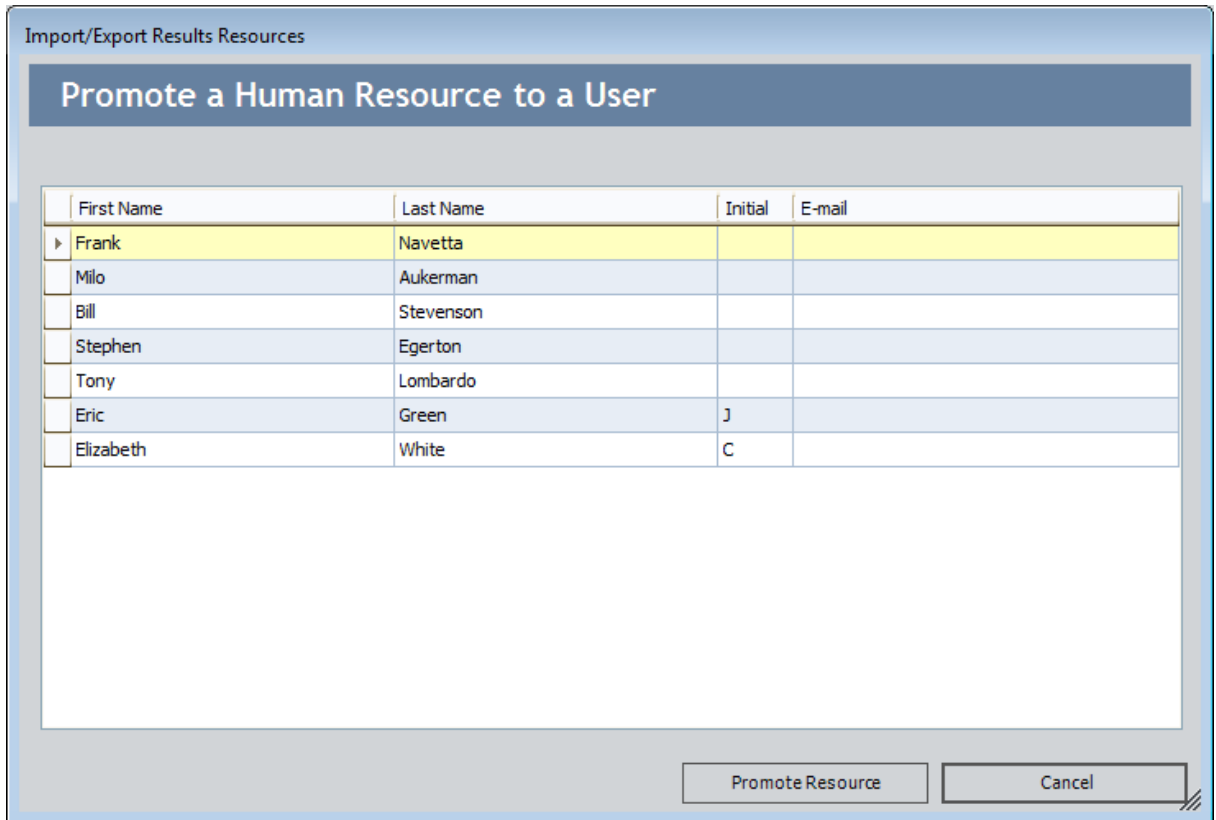
2. Highlight the row containing the Human Resource record that you want to promote, and click the **Promote Resource** button.

   The **Enter User Details** dialog box appears.

3. In the **User ID** text box, type a user ID for the new Security User.

   > **Note:** The LDAP User ID will be set automatically to the value that you for the Meridium APM User ID. After the Security User record has been created, you can open the record and modify the LDAP User ID if needed.

4. In the **Password** text box, type the password that is associated with the new Security User.

   > **Note:** When promoting a Human Resource record to a Security User record, you *must* specify a password. If you want the user to have a blank password, after the Security User record has been created, you can open the record and remove the password.

5. Select the **Super User?** check box if you want to designate the Security User as a Super User.

> 🛈 **Note:** The **Super User?** check box is enabled only if you are a Super User. Only Super Users can create other Super Users.

6. Click **OK**.

   A message appears, indicating that a Security User record was successfully created for the Human Resource record.

7. Click **OK** to close the message.

   The **Application Users** window returns to focus, displaying the promoted Security User in the list.

Promoting Human Resource from the APM Framework

Super Users can promote Human Resource records to Security User records via the MI Human Resource datasheet. Doing so will create a Security User that will be able to log in to the Meridium APM applications.

To promote a Human Resource record to a Security User record:

1. In the Meridium APM Framework, open the Human Resource record that you want to promote.

2. On the MI Human Resource datasheet, click the **Promote Resource** button.

   The **Enter User Details** dialog box appears.

3. In the **User ID** text box, type a user ID for the new Security User.

   > ⓘ **Note:** The LDAP User ID will be set automatically to the value that you supply for the Meridium APM User ID. After the Security User record has been created, you can modify the LDAP User ID, if needed.

4. In the **Password** text box, type the password that is associated with the new Security User.

   > ⓘ **Note:** When promoting a Human Resource record to a Security User record, you *must* specify a password. If you want the user to have a blank password, after the Security User record has been created, you can open the record and remove the password.

5. Select the **Super User?** check box if you want to designate the Security User as a Super User.

6. Click **OK**.

   A message appears, indicating that a Security User record was successfully created for the Human Resource record.

7. Click **OK** to close the message.

After the Security User record has been created, it can be viewed and modified in the Record Manager just like any other record. For more information on defining fields in Security User records, see the Configuration Manager Help. Note that the Configuration Manager application serves as the primary tool for managing Security User records. Via the Configuration Manager application, you can define the attributes of each Security User record and manage Security Group assignments and family-level privileges. Keep in mind that while Human Resource records can be promoted to Security User records via the Meridium APM Framework application, if you create a non-Super User, you will need to grant family-level privileges via the Configuration Manager application before that user will be able to perform tasks in the Meridium APM Framework application.

Configuring Security Group Membership

When you are [modifying an existing Security User](#) on the **Application User** dialog box, you can click the **Groups** tab to view the [Security Groups](#) to which the Security User is currently assigned and, if desired, modify those assignments.

To modify the list of Security Groups to which the Security User is assigned:

1. In the Configuration Manager, on the **Groups** tab of the [**Application User** dialog box](#), click the **Modify** link, which appears near the bottom of the window.

   The **Select Groups for [User]** dialog box appears.

   The **Select Groups for [User]** dialog box displays a hierarchy of [Security Groups that have been defined for your system](#). A check box appears to the left of each Security Group name. For any Security Group of which the Security User is currently a member, this check box is selected.

2. Select the check box next to any Security Group name to make the Security User a member of that Security Group.

3. If desired, you can clear any check box to remove the Security User from that Security Group.

4. When you are finished, click **OK**.

   The Security Group assignments are saved to the Security User account.

About the Query Time-out Limit

The *query time-out limit* is the amount of time in minutes the Meridium APM system will allow a new or modified query to attempt to return results before timing out. This setting allows you to control the performance of the queries in your system by enforcing a requirement that they meet a specific performance goal. This setting is used in the query design when a Security User whose query privilege setting is *Restricted By Timeout Limit* tries to save a new or modified query.

Before a Security User whose query privilege is *Restricted By Timeout Limit* can save a new or modified query, they will have to run the query so that the Meridium APM system can determine if it runs within the time-out limit. Otherwise, the save options will remain disabled. If a query time-out limit has been specified in that database, when the query runs, the Meridium APM system will allow the query to run until the specified query time-out limit has been met. After the time-out 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.

Specifying the Query Time-out Limit

The following instructions provide details on specifying the query time-out limit for Security Users who are restricted by query time-outs. These instructions do not provide a recommendation on choosing the time-out value itself. The value that you choose should match the amount of time that your organization has defined as a reasonable amount of time for a query to run before returning results. The value you enter must be a whole number (i.e., not a decimal value).

To specify the query time-out limit for Security Users who are restricted by the time-out limit:

1. In the Configuration Manager, click the **Tools** menu, point to **Global Preferences**, and then click **Query Timeout Limit**.

   The **Query Timeout Limit** dialog box appears.
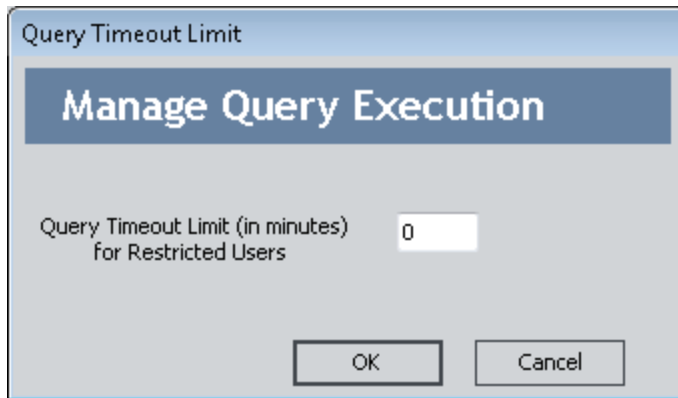
2. In the **Query Timeout Limit (in minutes) for Restricted Users** text box, replace the default value (zero) with a number that represents amount of time to which you want to limit queries for displaying results.

3. Click **OK**.

   The **Query Timeout Limit** dialog box closes, and your query timeout limit is saved.

About Managing Security Groups

A Security Group is a group of [Meridium APM Security Users](#) who share similar responsibilities or perform similar tasks in Meridium APM. After you create a Security Group, you can assign Security Users to the Security Group. Any Security User who is a member of a given Security Group will be granted the privileges defined for that Security Group and any groups. Security Groups can streamline the assignment of Security User privileges and help you organize Security Users according to their role in the system.

> ⓘ **Note:** Each Security User must be a member of at least one Security Group. By default, all users belong to the Everyone Security Group.

Security Groups serve two main purposes:

- They can have *functional privileges,* which control members' access to certain features in the system.
- They can have *[family-level privileges](#)* associated with them so that you can assign the same privileges to a group of similar Security Users.

Some of the [Security Groups that are included in the baseline Meridium APM database](#) have specific *functional* privileges associated with them that control access to certain features of the system. For example, members of the MI PROACT Administrator Security Group will have access to the Administrative Tools in RCA. Any user who is not a member of the MI PROACT Administrator Security Group will not be able to access the RCA Administrative Tools.

> ⓘ **Note:** Functional privileges are typically defined in the Meridium APM code and cannot be modified.

Family-level privileges determine each members' ability to access *data*. Family-level privileges are provided for many of the [baseline Meridium APM Security Groups](#) and can also be defined for any Security Groups that you create. Family-level privileges that are associated with baseline Security Groups can be modified.

Note that family-level privileges are spread down from Security Groups to Security subgroups. A Security Group should be given the *lowest* level of privileges allowed for any single member of that group. You can expand Security User privileges for individual Security Group members, but you cannot revoke from a Security User the privileges that are granted through any of its Security Groups. The more role-specific and task-specific you make your Security Groups, the easier it will be to define privileges for all of its members.

Each Security Group that you create in the Configuration Manager will have its own Group Home Page in the Meridium APM Framework application. The Home Page can serve as the starting point for common tasks performed by the Security Group's members. In addition, a Meridium APM Web Framework Home Page can be configured for any Security Group.

About Baseline Security Groups

Baseline Security Groups are provided so that you can grant a collection of security privileges that will enable certain a set of baseline functionality for users. For example, assigning a Security User to the MI RCM User Security Group will grant that user the permissions that are necessary to perform certain tasks in RCM.

> 🗋 **Note:** Before you can use the baseline Security Groups, you must first create Security Users. You can then assign any Security User to any Security Group to grant the user that set of privileges.

Keep in mind that assigning a user to a baseline Security Group does not necessarily grant that user ALL the permissions that are needed to perform all functions in a given module. In some case, you will need to assign users to a combination of Security Groups to give them all the necessary privileges. Note also that permissions to any customer-created entity or relationship family (i.e., any family not included in the baseline Meridium APM database) must be configured manually in addition to the baseline group assignments.

> ⓘ **Hint:** Rather than assigning privileges to custom families on a user-by-user basis, you can create your own Security Groups that have the custom privileges for a given module and then assign users to those groups. For example, you might create the RCM Custom User group that has privileges to RCM-related families. Then, by assigning a user to the MI RCM User and RCM Custom User groups, you could grant that user all the privileges needed to use RCM.

About the Everyone Security Group

The Everyone Security Group is included in the baseline Meridium APM database. Whenever you create a new Security User in the Configuration Manager, that user will be assigned automatically to the Everyone Security Group. While membership in the Everyone Group is not required (i.e., Security Users can be removed from this Security Group), *we recommend that you accept this default group assignment and keep all Security Users assigned to the Everyone Security Group.* Membership in the Everyone Security Group meets the basic requirements for accessing the Meridium APM system and provides users with View-level privileges to the APM Foundation families (e.g., Equipment and Functional Location).

By default, members of the Everyone Security Group have View privileges on the following entity families:

- Equipment
- Finding
- Functional Location
- Human Resource
- Inspection
- Inspection Profile
- Inspection Team Member
- Observation
- Personnel Certification
- Recommendation
- Reference Document
- Resource Role
- Security Group
- Security User
- Taxonomy References
- Work History
- Work History Detail

Members of the Everyone Security Group also have View privileges on the following relationship families:

- Equipment Has Equipment
- Functional Location Has Equipment
- Functional Location Has Functional Location(s)
- Group Assignment
- Has Certifications

- Has Event Detail
- Has Findings
- Has Inspection Profile
- Has Inspections
- Has Observations
- Has Reference Documents
- Has Roles
- Has Sub-Inspection
- Has Taxonomy Hierarchy Element
- Has Taxonomy Mapping
- Has Team Member
- Has Work History
- Is a User
- User Assignment

About the MI Catalog Administrator Security Group

Members of the MI Catalog Administrator Security Group have full access to all folders in the Catalog, including all Personal folders as well as the Public folder and the Baseline folder. This means that users can view, execute, modify, copy, add, and delete items in any Personal or Public folder. They can view, execute, and copy any item in the Baseline folder.

> ⓘ **Note:** Items in the Baseline folder cannot be modified, deleted, or added by any user, including Super Users and members of the MI Catalog Administrator Security Group.

Note that in order to work with items in any folder, members of the MI Catalog Administrator Security Group will need the necessary family-level privileges to any families involved in those items. For example, to execute a query on the Equipment family, members of the MI Catalog Administrator Security Group would need at least View privileges on the Equipment family.

About the Security Group Hierarchy

You can construct a Security Group hierarchy in the same way as you construct the entity family hierarchy. The advantage of constructing a hierarchy of Security Groups is that privileges are spread from higher-level Security Groups to lower-level Security Groups.

Consider the following hierarchy:

Pump Record Viewer

    Pump Record Updater

        Pump Record Administrator

To the **Pump Record Viewer** Security Group, you might assign View privileges to the Pump family. These privileges would automatically be spread down to the **Pump Record Updater** Security Group.

To the **Pump Record Updater** Security Group you could assign Update privileges to the Pump family, thereby giving members of the **Pump Record Updater** Security Group permission *both* to view *and* update Pump records. These privileges would spread down to the **Pump Record Administrator** Security Group.

To the **Pump Record Administrator** Security Group, you might grant Insert and Delete privileges, thereby giving members of the **Pump Record Administrator** Security Group *full* access to the Pump family.

You can construct the Security Group hierarchy however you like, with as many levels as you need. Keep in mind that the *lowest* levels in the hierarchy should be used for groups who will be granted the *most* privileges.

Overview of Role Security Groups

A *Role* is a special type of Security Group that grants and restricts access for its members to certain features in the system. All Role Security Groups have associated with them some Configuration Manager privileges. In other words, all members of a Role Security Group will be able to access the Configuration Manager in some way. Therefore, you will want to use these Security Groups to control the privileges of users who have access to the Meridium APM Administrative Applications and who need to perform tasks in the Configuration Manager.

> ⓘ Note: Super User privileges are unaffected by the introduction of Role Security Groups in V3.1.0. Super Users continue to have access to ALL Administrative Applications and Meridium APM Framework and Meridium APM Web Framework functions regardless of group membership. The difference is that whereas *only* Super Users could log in to the Configuration Manager previously, now you can provide limited access to the Configuration Manager via Role Security Groups.

Three Role Security Groups are delivered with the baseline Meridium APM product, each of which has specific privileges associated with it:

- **MI Configuration Role:** For users who should be allowed access to all the features in the Configuration Manager except security features.
- **MI Power User Role:** For users who should be allowed to import and export metadata in the Configuration Manager and who should be able to log in to the Meridium APM Framework application and the Meridium APM Web Framework.
- **MI Security Role:** For users who should be allowed to manage security in the Configuration Manager.

> ⓘ Note: Permissions associated with the baseline Role Security Groups are hardcoded and cannot be modified. If you want to use these groups, you must use them exactly as they are delivered. You cannot create your own Role Security Groups.

Of these three groups, only members of the MI Power User Role Security Group will be able to log in to the Meridium APM Framework application or the Meridium APM Web Framework. Members of the MI Configuration Role Security Group and members of the MI Security Role Security Group will be allowed to log in to the Configuration Manager only and will have restricted access to the features therein. Members of the MI Power User Role Security Group will also be able to log in to the Configuration Manager with restricted access.

You will want to make users members of only ONE of these Security Groups. If a user is assigned to more than one Role Security Group, where one Role *restricts* access to a feature and one role *grants* access to the same feature, the user will not be able to access that feature. Because users will be given the most restrictive privileges defined through Role Security Group membership, there is no advantage of assigning users to multiple Role Security Groups.

For members of the MI Power User Role Security Group, except for privileges that are dictated explicitly by the [privileges associated with the MI Power User Role Security Group](), access to Meridium APM Framework and Meridium APM Web Framework features is determined by membership in other Security Groups and by family-level privileges. In other words, you can give members of the MI Power User Role Security Group additional privileges by assigning them to additional Security Groups, with the exception of other Role Security Groups.

Privileges Associated with the MI Configuration Role

Members of the MI Configuration Role Security Group:

- Can view, create, update, and delete the following in the Configuration Manager:
  - Entity families.
  - Family fields, field properties, and field rules.
  - Relationship families and relationship definitions.
  - Datasheets.

  Note: These privileges are granted automatically through membership in the MI Configuration Role Security Group. No additional family-level privileges are needed to manage families.

- Have full access to the following features in the Configuration Manager:
  - ID Templates.

  Note: Permission to access data in existing records is granted for the purposes of implementing a new ID Template.

- The manage physical storage utility.

  Note: Permission to access data in the database is granted for the purpose of dropping families.

- System codes and tables.
- Units of Measure and Conversions.
- The Rules Library.
- The Compile Family feature.
- Data filters.
- The Usage Metrics Tracking configuration settings.
- Can access to the Catalog in the Import/Export Metadata tool in the Configuration Manager. Members of this group do not have access to records and links or security items.
- Have full access to the URL Manager application.

Members of the MI Configuration Role Security Group cannot:

- Access any security-related features in the Configuration Manager, including Security Users, Security Groups, and family-level privileges.
- Log in to the Meridium APM Framework application.

Privileges Associated with the MI Power User Role

Members of the MI Power User Role Security Group:

- Have access to the Import/Export Metadata tool in the Configuration Manager to import and export records and links, where access to data is limited by the user's family-level privileges.

  > ⓘ **Note:** Members of the MI Power User Role Security Group cannot perform any other functions in the Configuration Manager.

- Can log in to the Meridium APM Framework application and perform all tasks associated with that user's other Security Group assignments and family privileges.
- Have full access to all Personal, Public, and Baseline folders in the Meridium APM Catalog.
- Can view any links on the **Associated Pages** menu that are designated as **Apply when user is a Super User or a member of the MI Power User Security Group**, as defined in the URL Manager.
- Can create and modify action queries (i.e., Append, Update, and Delete). Access to saved queries is limited by the user's Catalog permissions.
- Can view and manage the Home Page for any Security User or Security Group and, when using the Send To feature, send a link to any Personal Home Page or Group Home Page. User access to specific data is determined by family-level privileges.

Members of the MI Power User Role Security Group cannot:

- Log in to the URL Manager.
- Perform tasks in the Configuration Manager that are not granted explicitly by the MI Power User Role group (see previous list).

Privileges Associated with the MI Security Role

Members of the MI Security Role Security Group can:

- View, create, update, and delete Security Users and Security Groups, and manage group membership.

  > ⓘ **Note:** Members of the MI Security Role group have access to all Security User properties except the **Super User** flag. Only [Super Users](#) can create other Super Users.

- Manage all aspects of family privileges for Security Groups and Security Users.
- Access the Import/Export Metadata tool to import and export security items only (i.e., **Security Groups and Privileges** and **Users, Privileges and Assignments**).
- Access the following features: Manage Content Validation, Database Comparison Utility, and Compile All Rules.

Members of the MI Security Role Security Group cannot:

- Access most of the family-management features, including the Rules Library and creation and compilation of metadata.
- Access System Codes and tables.
- Access to the units of measure or the conversions.
- Perform other tasks in the Configuration Manager that are not granted explicitly by the MI Security Role group.
- Log in to the Meridium APM Framework application.
- Log in to the URL Manager.

Viewing Existing Security Groups

To view the Security Groups that exist in the Meridium APM database:

- In the Configuration Manager, on the Configuration Manager main menu, click **Security**, and then click **Groups**.

    The **Application Groups** window appears, displaying a list of all the Security Groups that exist in the database.
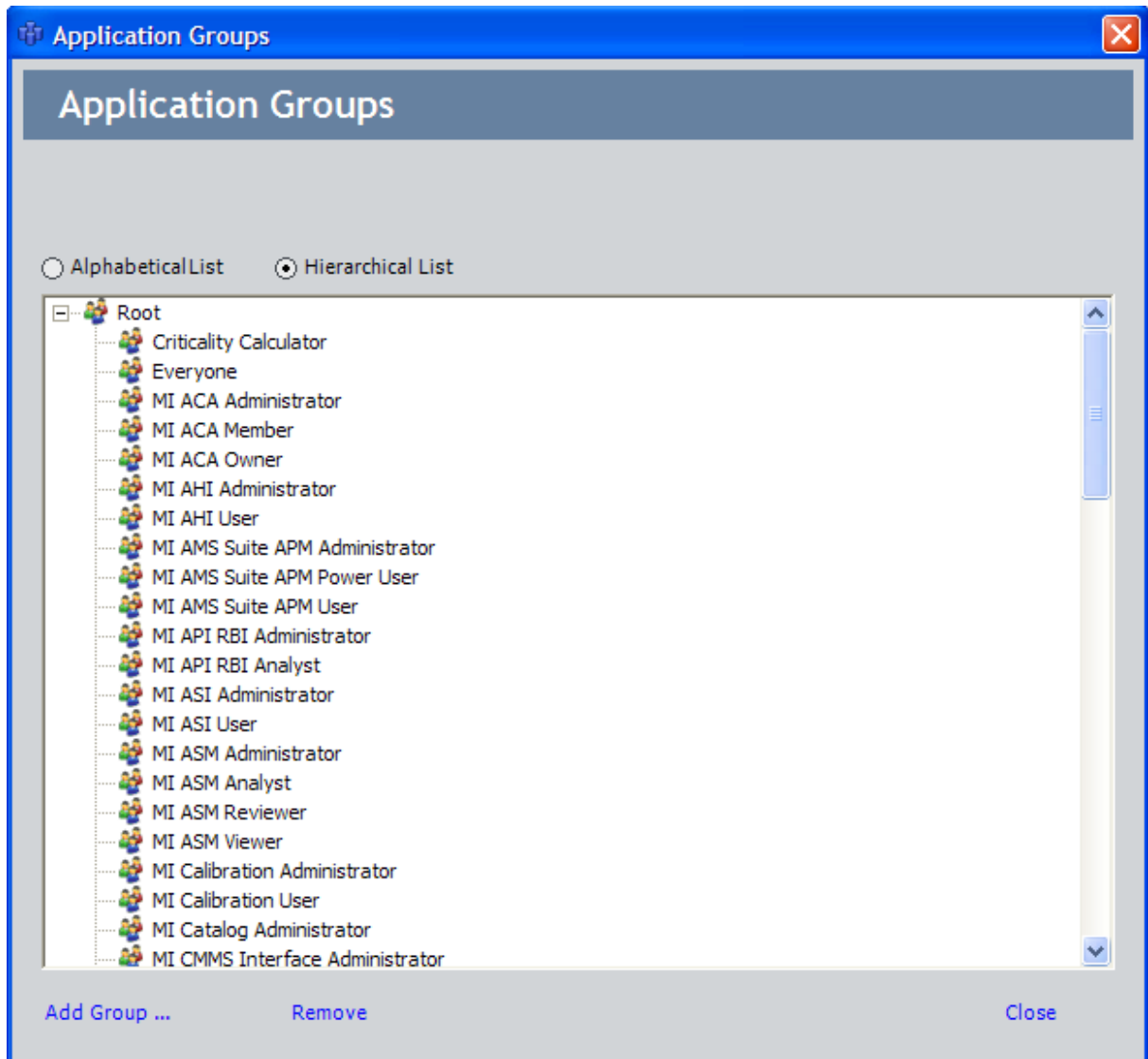
You can view Security Group names in alphabetical order or a hierarchical list by selecting the appropriate radio button. A hierarchical list consists of Security Groups in the database, starting with broad categories and branching off into more specific Security Groups as you select subfolders of the Root folder.

Creating New Security Groups

To create a new Security Group:

1. In the Configuration Manager, on the [Application Groups window](#), select the **Hierarchical** option at the top of the Security Group list, and then select the existing Security Group under which you want to add the new Security Group.

   > ⓘ **Note:** If you select the **Alphabetical** option, the new group will be added at the root level, regardless of which group is selected when you click the **Add Group** link.

2. Click the **Add Group** link.

   The **Application Group** window appears, displaying blank fields.

3. In the **Group ID** text box, type a unique ID to identify the Security Group within the system. You can construct the Group ID using any combination of characters. The Group ID cannot exceed 50 characters in length.

4. In the **Caption** text box, type a unique name for the Security Group. This name will appear to users to identify the Security Group throughout Meridium APM. The caption can be the same as the Group ID but must be unique with respect to other Security Group Captions. Note that you can manage translations for that string by clicking the 🌐 icon.

5. In the **Group Description** text box, type a description for the Security Group. This

field is optional.

6. For the **Group Status** field, accept the default selection: **Active**.

7. Click the **Save** button.

   The new Security Group is created and saved to the database.

After you create and save a Security Group:

- The **Edit Meridium APM Framework Home Page** link becomes enabled on the **Application Group** window. You can click this link to launch the Meridium APM Framework application and set up a Meridium APM Framework Home Page for the Security Group. For details on working with Home Pages, see the Meridium APM Framework Help. Note that you must click this link to allow members of the Security Group access to the Home Page. You are not required to add *content* to the Home Page, but you must *access* the Home Page so that it will be created and can then be accessed by others.

  > Note: All Super Users can modify the Meridium APM Framework Home Page for any Security Group. Members of the MI Security Role Security Group can modify any Security Group but cannot log in to the Meridium APM Framework application and, therefore, will be able to manage Group Home Pages via the Configuration Manager application only if they also belong to the MI Power User Role Security Group.

- The **Edit Meridium APM Web Framework Home Page** link becomes enabled on the **Application Group** window. You can click this link to open the **Meridium APM Web Framework Home Page** dialog box, where you can configure a Meridium APM Web Framework Home Page for the selected Security Group.

  > Note: Super Users and members of the MI Security Role Security Group can modify the Meridium APM Web Framework Home Page for any Security Group.

- You can add Security Users to the group.

Adding a Security User to an Existing Security Group

To add a Security User to an existing Security Group:

1. In the Configuration Manager, <u>on the **Application Groups** window</u>, in the list of Security Groups, double-click the Security Group to which you want to add a Security User.

   The **Application Group** window appears.

2. Below the **Assigned User(s)** list, click the **Add** link.

The **Users** dialog box appears, displaying the list of Security Users in the database.
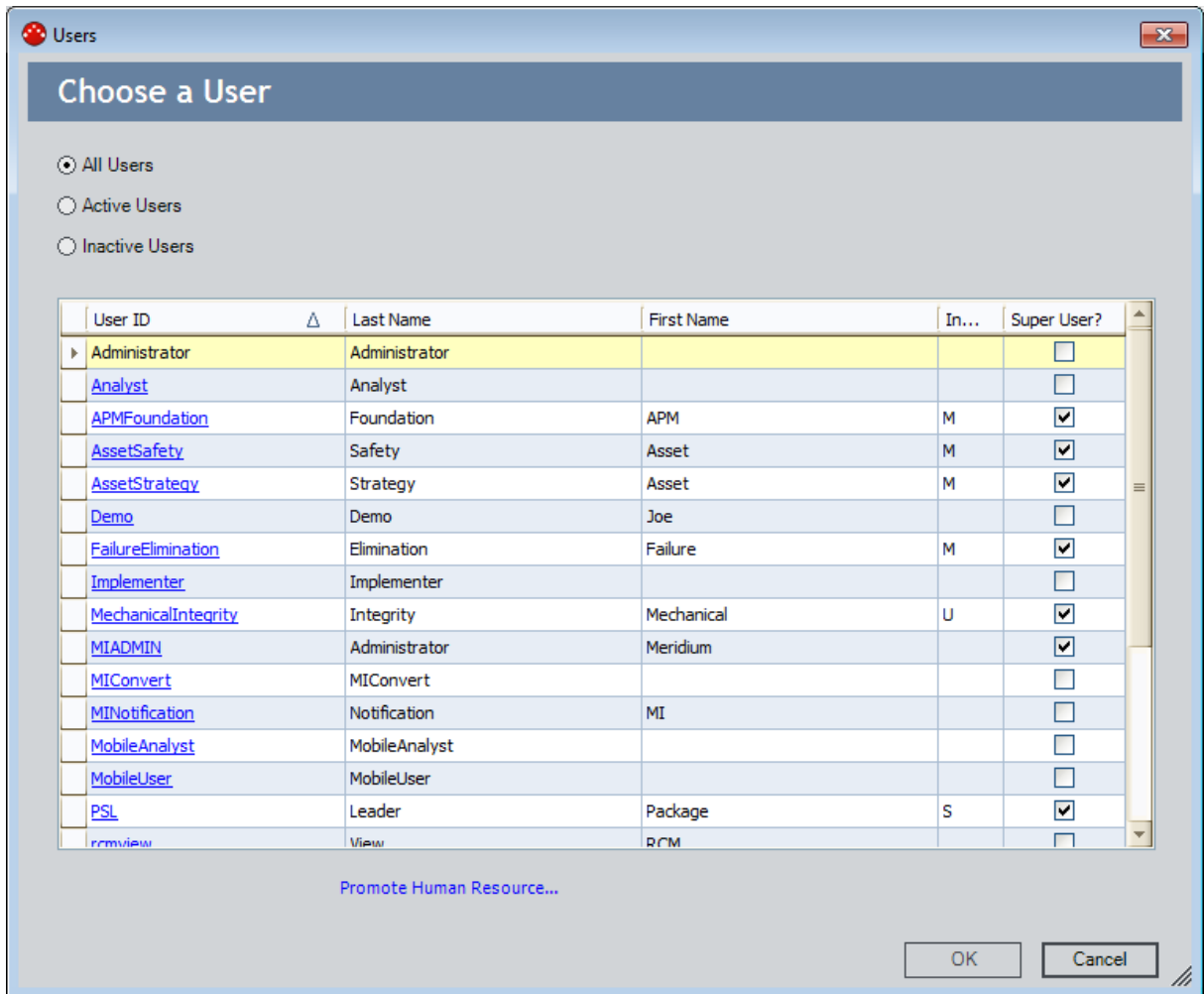
3. Select the row containing the Security User that you want to add.

   ⓘ **Hint:** You can use the Ctrl or Shift key to select multiple rows.

4. Click **OK**.

   If you selected more than 50 Security Users, a message will appear, indicating that the process of assigning those Security Users to the Security Group could take several minutes and asking if you want to proceed.

5. Click the **Yes** button to proceed with the operation. Clicking the **No** button will cancel the operation.

   The **Application Group** window returns to focus. The user(s) that you selected have been saved to the **Assigned User(s)** list.

> ⓘ **Note:** The **Save** button on **Application Group** window is disabled when you add Security Users to Security Groups. Clicking the **OK** button on the **Users** dialog box *will* save the Security User(s) to the selected Security Group.

Modifying Security Groups

To modify an existing Security Group:

1. In the Configuration Manager, on the **Application Groups** window, navigate the Security Group hierarchy to locate the Security Group that you want to modify.

2. Double-click the Security Group.

   The **Application Group** window appears, displaying the information that is currently associated with the Security Group.

3. Modify any of the current information. Note that:

- The Group ID is required and must be unique.
- The **Caption** is required and must be unique. It can be the same as the Group ID but must be unique with respect to other Security Group captions. Note that you can manage translations for that string by clicking the 🌐 icon.

4. If desired, click the **Edit Meridium APM Framework Home Page** link to launch the Meridium APM Framework application and set up a Home Page for the Security Group. For details on working with Home Pages, see the Help system in the Meridium APM Framework application.

> ⓘ **Note:** All Super Users can modify the Meridium APM Framework Home Page for any Security Group. Members of the MI Security Role Security Group can modify any Security Group but cannot log in to the Meridium APM Framework application and, therefore, will be able to manage Group Home Pages via the Configuration Manager application only if they also belong to the MI Power User Role Security Group.

5. If desired, click the **Edit Meridium APM Web Framework Home Page** link to open the **Meridium APM Web Framework Home Page** dialog box, where you can configure a Meridium APM Web Framework Home Page.

> ⓘ **Note:** Super Users and members of the MI Security Role Security Group can modify the Meridium APM Web Framework Home Page for any Security Group.

6. If desired, add Security Users to the group.

7. When you are finished modifying the Security Group, click the **Save** button.

Your changes are saved to the database.

Removing a Security User from a Security Group

To remove a Security User from a Security Group:

1. In the Configuration Manager, on the **Application Groups** window, in the list of Security Groups, double-click the Security Group from which you want to remove a Security User.

   The **Application Group** window appears, displaying the information that is currently associated with the Security Group.

2. In the **Assigned User(s)** list, select the Security User that you want to remove.

   > ⓘ **Hint:** You can use the Ctrl or Shift key to select multiple rows.

3. Click the **Remove** link.

   A confirmation message appears, asking if you really want to remove the Security User. If the Security User is the Home Page Administrator for the Security Group, the message also indicates that by removing the Security User, the Home Page will no longer have a Home Page Administrator.

4. Click the **Yes** button.

   The Security User is removed from the Security Group.

   > 🗎 **Note:** The **Save** button on **Application Group** window is disabled when you delete Security Users from Security Groups. Clicking the **Yes** button to confirm the deletion, *will* remove the selected Security User(s) from the Security Group.

Deleting Security Groups

> ⓘ **Note:** If a Security Group has been granted privileges to any family, when you try to delete the Security Group, an error message appears, and the Security Group will not be deleted. Before you delete a Security Group, you should first delete ALL of its family-level privileges. After doing so, you should be able to delete the Security Group successfully.

To delete a Security Group:

1. In the Configuration Manager, on the Application Groups window, in the list of Security Groups, highlight the group that you want to delete.

2. Click the **Remove** link.

   A confirmation message appears, asking if you really want to delete the Security Group.

3. Click the **Yes** button.

   The Security Group and all its subgroups are deleted from the database.

About Security Reports

Via the Configuration Manager application, you can create Security Groups, assign Security Users to those Security Groups, and assign family-level privileges to those Security Groups. Via the Meridium APM Framework application, you can grant Security Groups permission to access Public folders in the Meridium APM Catalog.

The tools provided in the Configuration Manager and Meridium APM Framework application let you view and manage permissions and assignments on a per-Security Group, per-family, and per-folder basis. To make it easier to view *all* the information associated with a given Security Group, Meridium APM provides a set of baseline reports that return comprehensive information for *all* the Security Groups in the database.

> Note: Any baseline item in the Baseline folder is also available in the corresponding Public folder. All Public folders exist by default. Throughout this documentation, we refer to items in the Public folder.

Security reports and their supporting queries are stored in subfolders within the Catalog folder \\Public\Meridium\Modules\Administrative Tools. In general, the information that is retrieved using these security queries and security reports is meant to be viewed using the *report*, which provides a formatted view of the query results. In order to view these reports, you must have SQL Server Reporting Services set up and properly configured.

If you want the reports to function as they do in the baseline database, you should not modify the name of the Catalog folder \\Public\Meridium\Modules\Administrative Tools\Security Queries, change the name or content of any items it contains, or move any item out of the folder. These security reports are built from the corresponding security queries. Modifying or moving any of the queries could cause the reports to stop functioning.

The topics in this section of the documentation provide details about these security queries and security reports. Note that these security reports and security queries are not run automatically by any Meridium APM module or feature and can be run like any other query or report in the Meridium APM Catalog.

Security Queries

The queries in the Catalog folder \\Public\Meridium\Modules\Administrative Tools\Security Queries exist to support a [report of the same name](#), which are stored in the Catalog folder \\Public\Meridium\Modules\Administrative Tools\Security Reports.

Security Reports

The following table provides a description of the security reports that exist in the Catalog folder \\Public\Meridium\Modules\Administrative Tools\Security Reports. Note that these reports are based upon queries of the same name, which are stored in the folder \\Public\Meridium\Modules\Administrative Tools\Security Queries.

Report	Supporting Query	Description
Family Security Assignments by Group ID	Family Security Assignments by Group ID	Returns a list of all the family-level privileges defined for each Security Group record in the database. The results are sorted by the Group ID field in the Security Group family so that you can see the family-level *privileges per Security Group*.
Group Security Assignments by Family ID	Group Security Assignments by Family ID	Returns a list of Security Groups and privileges assigned to each entity and relationship family in the database. The results are sorted by family ID so that you can see the *privileges per family*.
Group Security Assignments by User ID	Group Security Assignments by User ID	Returns a list of all the Security Group records in the database and information about the Security Users who are assigned to each group. The results are sorted first by Group ID and then by User ID so that you can see the Security User *assignments per Security Group*. This report returns the same information as the User Security Assignments by Group ID report but in a slightly different format.

Report	Supporting Query	Description
Public Folders Group Security Assignments	Public Folders Group Security Assignments	Returns a list of all the Security Groups in the database that have been granted privileges to a Public Catalog folder and information about the folder to which the group has been granted privileges. The results are sorted first by Group ID and then by folder ID so that you can see the *Catalog privileges per Security Group*.
User Security Assignments by Group ID	User Security Assignments by Group ID	Returns a list of all the Security Group records n the database and information about the Security Users who are assigned to each group. The results are sorted first by Group ID and then by User ID so that you can see the *Security User assignments per Security Group*. This report returns the same information as the Group Security Assignments by User ID report in a slightly different format.

Configuring Home Pages for the Meridium APM Framework Application

After a Security Group has been created and saved, on the **Application Group** dialog box, the **Edit Meridium APM Framework Home Page** link will be enabled. You can use this link to access the Meridium APM Framework application and configure the Meridium APM Framework Home Page for that Security Group. Note that members of the Security Group will not have access to the Home Page for the Security Group until you click the **Edit Meridium APM Framework Home Page** link, however you do not have to *modify* the Home Page to allow access to the Home Page.

To configure a Meridium APM Framework Home Page:

- In the Configuration Manager, on the **Application Group window**, click the **Edit Meridium APM Framework Home Page** link.

  The Meridium APM Framework application is launched. After you log in, you can set up a Home Page for the Security Group.

About Data Filters

The Meridium APM security model allows for dynamic *data filtering*, which lets you configure the system to limit a user's access to records, based on the information stored within the record fields. An administrative user can select a root-level entity family and then apply a Meta-SQL filter for each operation (Select, Insert, Update, or Delete) that members of a selected Security Group will be allowed to perform in the Meridium APM Framework application. By using data filters, you might, for example, implement site-level security. You can allow different Security Groups located at different sites within a single organization to perform only assigned operations on records belonging to specific sites.

Note that while data filters can be *defined* only for root-level entity families, the filters are *applied* automatically to each subfamily of the root-level family. Filter criteria itself can be designed such that it applies only to specific subfamilies within the family. Filters can be assigned to any Security Group and will be inherited by all the users of that Security Group, but filters are not spread from the Security Groups to Security subgroups. Data filters cannot be defined on relationship families.

The instructions in this documentation provide details on setting up data filters in the Configuration Manager so that Security Users will be able to view, edit, create, or delete only the records that are available to the Security Group to which they belong. Note that before you can set up data filters, you must first configure the Security Users and Security Groups to which the filters will apply.

Operations for Which Data Filters Can Be Applied

When you create a data filter, you must choose which operation will cause the filter to be applied. The criteria that you define will be applied when a user belonging to the selected Security Group performs the specified operation. You can choose from the following operations:

- **Select:** Filter criteria will be applied when a Select operation is performed. This means that the criteria will be applied when a user runs a Select query and that it will also be applied when a Select query is executed behind-the-scenes. For example, searches use Select queries to retrieve search results, and other components use Select queries to compile lists of records and display them to the user. Note that this type of filtering applies only to queries that select entity families. Filtering will not be applied against queries on physical database objects (e.g., tables or views).

- **Insert:** Filter criteria will be applied when a user belonging to the selected Security Group creates a new record. The filter criteria itself might prohibit the user from creating the record or set certain values automatically based on the user's group membership.

- **Update:** Filter criteria will be applied when a user belonging to the selected Security Group updates an existing record. The filter criteria itself might prohibit the user from modifying certain values based on the user's group membership.

- **Delete:** Filter criteria will be applied when a user belonging to the selected Security Group updates an existing record. The filter criteria itself might prohibit the user from deleting the record based on the user's group membership.

About Defining Criteria for Data Filters

Any valid Meta-SQL expression can be used as the criteria string for a filter. Field references must refer either to system fields or fields defined on the family itself that are marked as an ID field.

Beyond this, Meta-SQL expressions can be of any degree of complexity. The expression can include a sub-query, for example, or reference a query stored in the Catalog. The expressions may also include predefined or database functions, such as UserKey to TO_CHAR. For example, each of the following represents a valid Meta-SQL filter criteria:

- [ANALYSIS_PUBLIC_FLG] = 'r;Y' OR ENTY_KEY IN (SELECT t.[ANALYSIS_ENTY_KEY] FROM [MI Analysis Team] t WHERE t.[TEAM_MEMBER_SEUS_KEY] = UserKey()

- [SITE_ID] IN (! 'r;Public\System Queries\Corporate Sites')

- [REC_CREATE_DT] >= ADD_MONTHS(Now(), -24)

Defining a New Data Filter

Use the following instructions to select a family and assign filter criteria to selected Security Group(s). Note that you can define a Meta-SQL filter criteria on any field defined for a root-level family.

To define a new data filter:

1. In the Configuration Manager, in the **Entity Family/Relationship Family** pane, click the **Entity Family** tab.

2. Select the family for which you want to define the data filter.

   > ⓘ **Note:** Data filters can be defined only for root-level families. The Manage Data Filters link will be enabled only when you select a family for which data filters can be defined.

3. In the **Tasks** section, click the **Manage Data Filters** link.

   The **Family Filters** window appears.

4. In the first empty row, click the down arrow in the **Group** column, and select the Security Group for which you want to define the filter. Be sure to select the specific Security Group to which the filter should be applied as filters defined for groups are not spread to subgroups.

5. In the **Operation** column, select the operation that will invoke the filter: **Select**, **Insert**, **Update**, or **Delete**.

   > ⓘ **Note:** Each row represents a single data filter. Only one operation can be selected for a given data filter. If you want to apply the same criteria to multiple operations, you must create an individual data filter for each. You can, however, copy the criteria from one filter and paste it into the **Criteria** cell of another filter.

6. In the **Criteria** cell, type the desired filter criteria in the text field.

7. Click the **Test** button in the **Criteria** field to test the criteria and make sure that it is valid.

   A message appears, confirming whether or not the filter criteria are valid.

8. When you selected a Security Group in the first empty row, the system should have added a new empty row. If you want to create another data filter, repeat steps 4-7 in the next empty row. Continue in this way until you have defined all the desired data filters.

9. When you are finished defining data filters, click the **OK** button at the bottom of the **Family Filters** window.

   The filters are saved to the database.

Editing the Criteria for an Existing Data Filter

To modify the criteria for a data filter that already exists for a family:

1. In the Configuration Manager, in the **Entity Family/Relationship Family** pane, click the **Entity Family** tab.

2. Select the family for which you want to define the data filter.

3. In the **Tasks** section, click the **Manage Data Filters** link.

   The **Family Filters** window appears.

4. Locate the row containing the data filter that you want to modify.

5. Click the down arrow in the **Criteria** to display a text box.

6. Modify the criteria, as desired.

7. Click the **Test** button in the **Criteria** field to test the criteria and make sure that it is valid.

   A message appears, confirming whether or not the filter criteria are valid.

8. When you are finished modifying filter criteria, click the **OK** button at the bottom of the **Family Filters** window.

   Your changes are saved to the database.

Deleting an Existing Data Filter

To select a family and delete an existing filter:

1. In the Configuration Manager, in the **Entity Family/Relationship Family** pane, click the **Entity Family** tab.

2. Select the family for which you want to define the data filter.

3. In the **Tasks** section, click the **Manage Data Filters** link.

   The **Family Filters** window appears.

4. Select the row containing the filter that you want to delete.

5. Press the Delete key on your keyboard.

   A message appears, asking you to confirm the deletion.

6. Click the **Yes** button.

7. On the **Family Filters** window, click **OK** to close the window.

Introduction to Family-Level Privileges

Family-level privileges let you define which Security Users and Security Groups can access records associated with a given family. By setting up family-level security privileges, you can determine whether users will be able to view, modify, create, and delete records belonging to particular families. For example, a Security User with *Create* privileges on the Equipment family will be able to create new Equipment records, while users with *View* privileges on the Equipment family will be able only to search for and view existing Equipment records.

Family-level privileges can be defined for both entity families and relationship families. For each family, you can define View, Update, Insert, and Delete privileges, but the functionality associated with each level of privileges is slightly different for entity families and relationship families.

When assigning family-level privileges, you can assign privileges for individual Security Users or for entire Security Groups.

- When you assign family privileges at the Security *Group* level, ALL members of that Security Group will have those privileges.
- When you assign family privileges at the Security *User* level, *only* that specific Security User will have those privileges.

Assigning privileges at the Security Group level can be more efficient, as the privileges assigned to a Security Group will be inherited by every member of that group. Assigning privileges at the Security User level, however, gives you more flexibility in customizing privileges for individual Security Users.

Some family-level security privileges are configured for the baseline Security Groups to provide access to the baseline Meridium APM families. Others privileges will need to be configured manually before users will be able to access certain features in Meridium APM. In addition, you will need to configure privileges manually for families that you create.

Entity Family Privileges

For each entity family, you can grant users View, Updated, Insert, and Delete privileges.

- **View** privileges allow users to *view* records that belong to the family. Users with only View privileges on a family will be able to *open* existing records belonging to that family but would not be able to create, modify, or delete them.

  For typical users, View privileges are a pre-requisite to Update, Insert, and Delete privileges because they provide users with the initial access required for modify, create, and delete operations. For example, View privileges on a family would allow users to:

  - Search for records in that family and open them in the Record Manager.

  - Create a Select query on that family.

  - Open reports that include that family.

- **Update** privileges allow users to *modify* records that belong to the family. Users with only Update privileges on a family would be able to modify existing records belonging to that family but would not be able to view, delete, or create records in that family.

  Typically, you would not grant Update-only privileges to a user because while that user would be able to modify records in that family, without View privileges, they would not be able to search for them or open them in the Record Manager. Update-only privileges, however, may be used to support processes and operations that occur outside of the user interface (i.e., interfaces or scheduled jobs).

- **Insert** privileges allow users to *create* records in the family. Users with only Insert privileges on a family would be able to create records in that family but would not be able to view, modify, or delete them.

  Typically, you would not grant Insert-only privileges to a user because while that user would be able to create records in that family, without View privileges, they would not be able to initiate the record-creation process in the Meridium APM Framework (i.e., the family would not appear in the list on the **Select Family** dialog box). Insert-only privileges, however, may be used to support processes and operations that occur outside of the user interface (i.e., interfaces or scheduled jobs).

- **Delete** privileges allow users to *delete* records that belong to the family. Users with only Delete privileges on a family would be able to delete records belonging to a family but would not be able to view, modify or create them.

  Typically, you would not grant Delete-only privileges to a user because while that user would be able to delete records in that family, without View privileges, they would not be able to perform a search to find the records that they wanted to delete. Delete-only privileges, however, may be used to support processes and operations that occur outside of the user interface (i.e., interfaces or scheduled jobs).

Relationship Family Privileges

Relationship family privileges are similar to entity family privileges in that the same privilege level are available: View, Update, Insert, Delete. Each permission level, however, provides access that is slightly different from the access provided through entity family privileges.

Remember that relationship families are used to create links between records in entity families. Consider an example where the *Equipment* family is related to the *Failure* family through the *Has Failure* relationship. In this case, to provide a user with full access to an Equipment record and its associated Failure record, that user would need privileges to three families: Equipment, Failure, and Has Failure.

> ℹ **Note:** Privileges on a relationship family do not automatically provides access to the predecessor and successor families; explicit entity family privileges are require for that.

For relationship families:

- **View** privileges allow users basic access to the links that relate predecessor and successor records. Users with View-only access to a relationship family will be able to open existing linked records but will not be able to link and unlink records using that relationship. Note that...
    - To access linked records using the Master/Detail datasheet, users *must* have privileges on the master family, the detail family, and the relationship family.
    - To access linked records using other types of datasheets, privileges are not required on the relationship family, provided that the user has at least View-level privileges on both the predecessor and successor entity families.

- **Update** privileges allow users to *modify* existing links in a relationship family. This applies only in cases where fields are defined for the relationship family. Users with Update privileges will also need View privileges.

- **Insert** privileges allow users to *link* records together using that relationship family. Using the previous example, a user would need Insert privileges on the Has Failure family to link Equipment records to Failure records. Users with Insert privileges will also need View privileges.

- **Delete** privileges allow users to *unlink* records associated with the relationship family. Using the previous example, a user would need Delete privileges on the Has Failure family to *unlink* Failure records from Equipment records. Users with Insert privileges will also need View privileges.

About the Inheritance of Family-Level Privileges

The privileges assigned for a family are automatically inherited by its subfamilies. For instance, consider the following hierarchy:
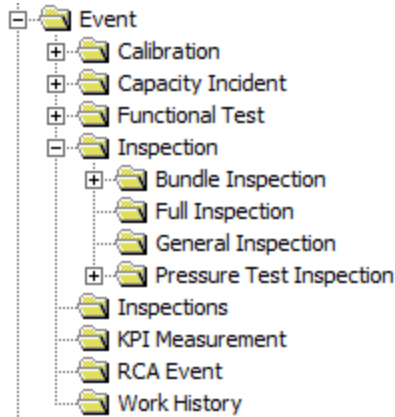
In this example, any privileges defined for the Event family will also apply to the Calibration and Inspection families (and so on). Similarly, any privileges defined for the Inspection family will also apply to the Bundle Inspection and Full Inspection families (and so on).

When you view the **Family Privileges** window for a subfamily, any privileges that have been inherited from a higher-level family are displayed as read-only fields. For example, in the hierarchy shown above, if privileges were assigned for the Event family, when you accessed the privileges for the Calibration family, the **Family Privileges** window might look like the following image, where the privileges defined at the Event level are read-only.
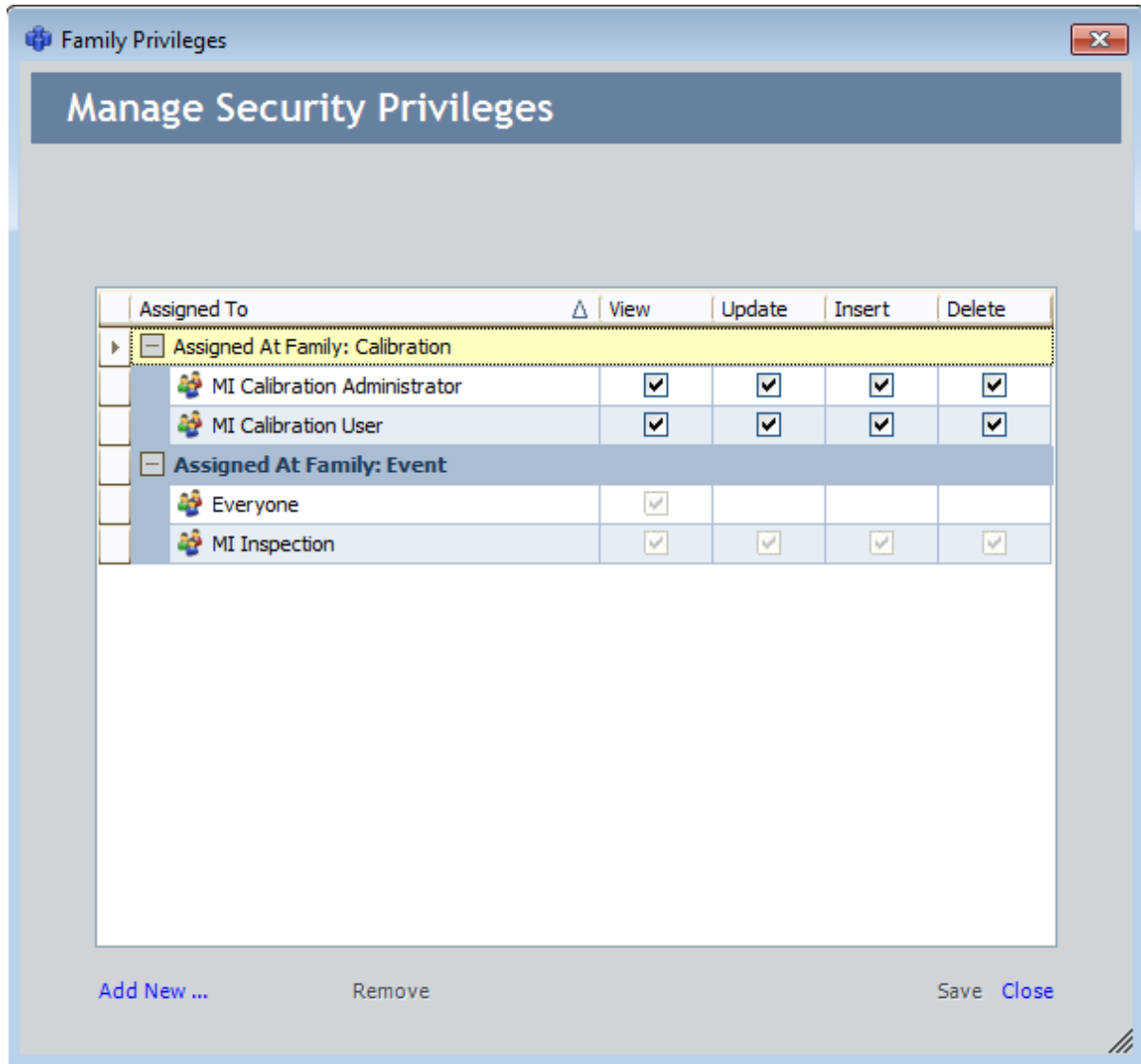
The name of the family from which each privilege has been inherited is indicated in the **Assigned To** column. In this example, privileges for the Everyone Security Group are defined at the Event level. If desired, you can define additional privileges directly to the Calibration family that will apply only to that family. Inherited privileges can be modified only on the family for which they are directly defined. For instance, to modify the privileges for the Everyone Security Group in this example, you would need to access the **Family Privileges** window for the *Event* family.

Note that in various places throughout Meridium APM, families are displayed using a hierarchical view. For example, in the Meridium APM Framework application, the Browser provides a hierarchical view of families, which you can use to access records. In cases like this, users must have at least *View* privileges at the highest level in the hierarchy in order to access any subfamily within that branch of the tree. For instance, in the example shown above, users would need View privileges on the *Event* family in order to

see the Calibration family in the Browser. Because of inheritance, this would allow users to view ANY record within the Event family hierarchy.

The Cumulative Effect of Family-Level Privileges

You have the option of defining family-level privileges for both Security Users and Security Groups. Because each type of privileges offers different advantages, you will probably want to use a combination of both Security User and Security Group privileges to achieve the specific privileges that are needed for your system. Note, however that family-level privileges are *cumulative*. A given Security User will be granted the sum of all privileges assigned to that Security User and to all the Security Groups of which that Security User is a member. In addition, note that privileges granted to Security Groups spread down automatically to all of their security subgroups.

Consider the following example, which shows the privileges defined for the Inspection Recommendation family.

The first row in the list of privileges indicates that John Smith has View privileges on the Inspection Recommendation family. In the next section, which shows the privileges defined at the Recommendation level, among other Security Groups, the MI Inspection Security Group has View, Update, Insert, and Delete privileges on the Recommendation family.

Now, assume that John Smith is a member of the MI Inspection Security Group. In this case, John Smith will have ALL privileges on the Recommendation family. The privileges assigned to the MI Inspection group are added to the privileges assigned to John Smith at the Security User level, therefore giving him full privileges to the Recommendation family.

Due to the cumulative effect of family-level privileges you will want to assign to a given Security Group the *lowest* level of privileges that you want to grant to any member of that Security Group. Then, you should grant additional privileges to individual Security Users who need to have more privileges.

Viewing the Privileges Currently Assigned for Families

You can view the current privileges for a family to understand the rights assigned to that family for a Security Group or a Security User and to decide if the Security Group or User should be granted additional privileges or have privileges taken away.

To view the privileges currently assigned to a family:

1. In the Configuration Manager, in the **Entity Family/Relationship Family** pane, click the **Entity Family** tab (if you want to view privileges for an entity family) or the **Relationship Family** tab (if you want to view privileges for a relationship family).

2. Select the family whose privileges you want to view.

3. In the **Tasks**section, click the **Manage Security Permissions** link.

   The **Family Privileges** window appears, displaying a list of all the Security Users and Groups who currently have access to the family and its associated records. You can edit the privileges by clicking the **View**, **Update**, **Insert**, or **Delete** check box next to the Security User or Group.

Family Privileges

Manage Security Privileges

Assigned To	View	Update	Insert	Delete
▷ ⊟ Assigned At Family: Active Strategy				
AQA Secured Super Users	☑	☑	☑	☑
MI ASI User	☑	☐	☐	☐
MI ASM Administrator	☑	☐	☐	☐
MI ASM Analyst	☑	☑	☑	☑
MI ASM Reviewer	☑	☑	☑	☑
MI ASM Viewer	☑	☐	☐	☐

Add New ...   Remove   Save   Close

Granting Family Privileges

To add security rights to a family:

1. In the Configuration Manager, access the **Family Privileges** window for the family whose privileges you want to modify.

2. Below the list of users and groups, click the **Add New** link.

    The **Add New Privilege** dialog box appears.
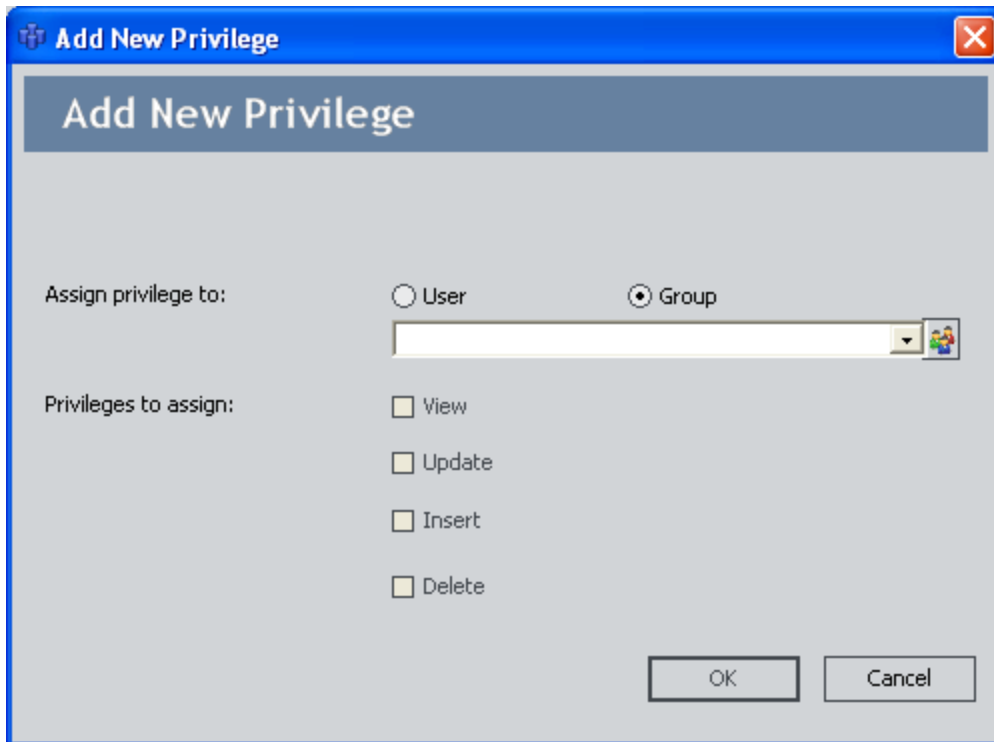
3. Click the **User** or **Group** option, depending on which type of privilege you want to add.

4. In the list that appears below the options, select the desired Security User or Group.

    > 🛈 **Note:** If you selected the **User** option, this list will contain all the Security User accounts that exist in the system. If you selected the **Group** option, the list will contain the Security Groups that exist in the system.

5. In the **Privileges to assign** area, select the check box(es) next to the type of privileges you want to assign to the selected Security User or Group.

    -   **View:** The Security User or Group will be able to view records that belong to the family.
    -   **Update:** The Security User or Group will be able to update records that

belong to the family and any report, query, graph, or other Meridium APM item that uses the family.

- **Insert:** The Security User or Group will be able to create records in the family.

- **Delete:** The Security User or Group will be able to delete records that belong to the family and any report, query, graph, or other Meridium APM item that uses the family.

6. Click **OK**.

The privileges are saved to the family.

Editing Family Privileges

You can edit family privileges for a root-level family only. This means that if you access the **Family Privileges** window for a non-root-level family, the check boxes indicating the current privileges will be disabled.

To change the security rights of a family:

1. In the Configuration Manager, access the **Family Privileges** window for the family whose privileges you want to modify.

2. In the list of Security Users and Groups who currently have rights to the family, locate the desired Security User or Group.

3. In the row for that Security User or Group, select or clear any of the check boxes:

   - **View:** When selected, gives the Security User or Group permission to view records that belong to the family.

   - **Update:** When selected, gives the Security User or Group permission to update records that belong to the family and any report, query, graph, or other Meridium APM item that uses the family.

   - **Insert:** When selected, gives the Security User or Group permission to create records in the family.

   - **Delete:** When selected, gives the Security User or Group permission to delete records that belong to the family and any report, query, graph, or other Meridium APM item that uses the family.

4. When you are finished, click the **Save** link.

   Your changes are saved to the system.

Removing Family Privileges

You can remove family privileges for a root-level family only. This means that if you are viewing the **Family Privileges** window for a non-root-level family, such as the Inspection Recommendation family, the **Remove** link will be disabled.

To remove security rights when a user or group no longer needs access to a family:

1. In the Configuration Manager, access the **Family Privileges** window for the family whose privileges you want to modify.

2. In the list of Security Users and Groups who currently have rights to the family, select the row containing the Security User or Group that you want to delete.

3. Click the **Remove** link.

   A confirmation message appears, asking if you really want to remove the Security User or Group.

4. Click the **Yes** button.

5. Click the **Close** button when you have finished modifying the family privileges.

What Is LDAP Integration?

Through its support for the Lightweight Directory Access Protocol (LDAP), Meridium APM gives you the option of integrating your Meridium APM system with Microsoft Active Directory, which can serve as the master repository for storing information about Meridium APM users. The Meridium APM LDAP integration consists of two main components:

- **Authentication:** Users will be authenticated against their Microsoft Active Directory credentials and logged in to Meridium APM automatically when they launch any Meridium APM application that supports automatic login, without having to supply a user ID or password. This method of automatic login offers two advantages compared to the non-LDAP automatic login method:
    - It provides increased security, as the user's Meridium APM user status will be in sync with their Active Directory status.
    - It does not require that users' Active Directory user names match their Meridium APM User IDs.

  > **Note:** Meridium APM provides a non-LDAP automatic login feature, through which Meridium APM users are authenticated against the information that they use to log in to their Windows operating systems. If their Meridium APM User ID matches the user name that they use to log in to their Windows operating system, they will be logged in to Meridium APM automatically whenever they launch a Meridium APM application that supports automatic login. This automatic login feature is enabled automatically on all Meridium APM systems and will be available if LDAP integration has not been enabled.

- [Synchronization](): User information in Meridium APM can be synchronized with the information in Microsoft Active Directory so that updates made in Active Directory are reflected in Meridium APM Security User records and Human Resource records. Additionally, new Security Users will be created automatically as users are added to [Active Directory groups that have been identified as Meridium APM groups](), and Meridium APM Security Users can be added to and removed from Meridium APM Security Groups based upon their membership in associated Active Directory groups.

  Alternatively, synchronization can be set to occur based on the LDAP User ID instead of the Meridium User ID. This option should only be used in specific situations where the Meridium User ID differs from the LDAP User ID.

LDAP integration is an optional feature, but because of its benefits, we strongly recommend that you configure your system for LDAP integration. To use the Meridium APM LDAP integration:

- You must have a Microsoft Active Directory system that stores information about users who will need to access Meridium APM. The Meridium APM documentation does not provide details on setting up and configuring your Active Directory system. For details on performing these tasks, consult your Microsoft

documentation.

- [You will need to perform various tasks in Meridium APM to enable and configure LDAP integration.](#)

LDAP Integration Data Model

The following image shows the entity families and relationship families that are used in Meridium APM for LDAP integration. This image provides a visual representation of the records and links that you will need to create via the Meridium APM Framework application to enable LDAP integration.
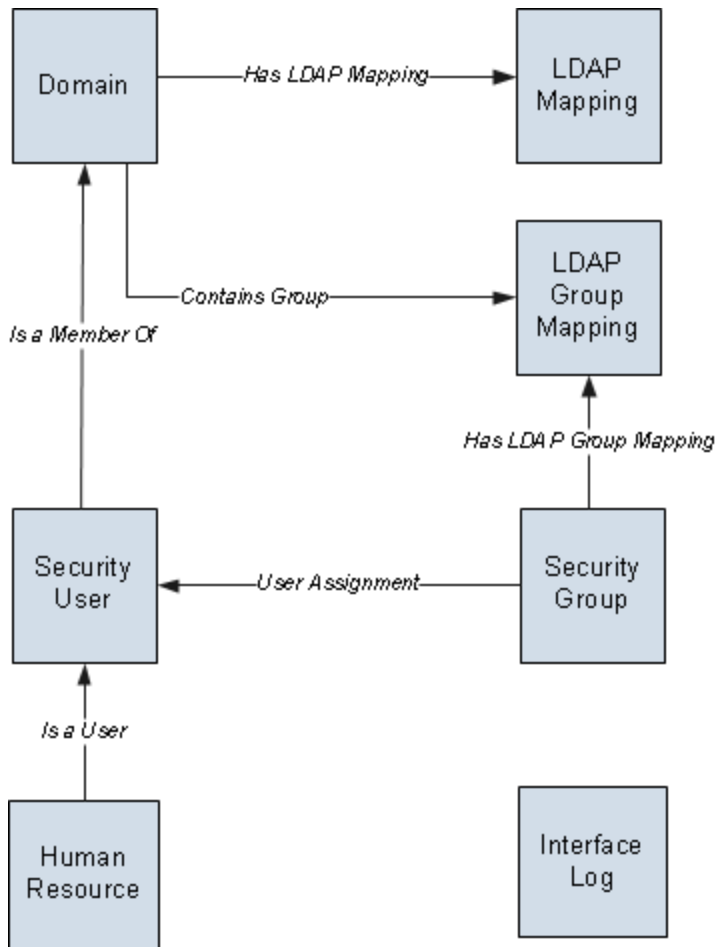
The following table provides a description of how the records and links in the preceding illustration are used by the Meridium APM LDAP integration feature.

Record/Link Type	Description
Domain	Stores information about the Microsoft Active Directory domains that you want to use for LDAP integration. Domain records contain the information that allows Meridium APM to connect to an Active Directory domain and details that allow Meridium APM to identify the users and groups within that domain who should be included in LDAP authentication and synchronization operations.

Record/Link Type	Description
Is a Member Of	Links each Security User record to a Domain record to identify the Active Directory domain in which the corresponding Active Directory user resides.
Security User	Stores the identifying information for Meridium APM users. For LDAP integration to work properly, each Security User record must be linked to a Domain record. This link is created automatically during synchronization.
Is a User	Links Security User records to their corresponding Human Resource record.
Human Resource	Stores the identifying information for a given Meridium APM Security User. Whenever a new Security User record is created, either manually via the Configuration Manager application or automatically via the LDAP synchronization process, a corresponding Human Resource record is created automatically and linked to the Security User record. Each LDAP synchronization operation will update the Security User record and its corresponding Human Resource record with any updated information that is found in the corresponding Active Directory user account.
Has LDAP Mapping	Links Domain records to LDAP Mapping records.
LDAP Mapping	Defines how fields in Microsoft Active Directory user accounts correspond to fields in Meridium APM Security User records. The mappings that are defined in LDAP Mapping records are used to synchronize data between Active Directory and Meridium APM. The LDAP Mapping records tell Meridium APM what information should be retrieved from Active Directory and where it should be stored in Meridium APM.
LDAP Group Mapping	Identifies groups in a given Active Directory domain that should be included in the Meridium APM LDAP integration. Only members of groups defined in LDAP Mapping records and linked to Domain records will be included for authentication and synchronization. The Domain record to which an LDAP Group Mapping record is linked should contain a Group Filter that includes the Active Directory group identified in the LDAP Group Mapping record.
Contains Group	Links Domain records to LDAP Group Mapping records.

Record/Link Type	Description
Security Group	Stores information about Meridium APM Security Groups. Note that Meridium APM Security Groups are not created or updated automatically through LDAP integration. Only group membership is affected by the synchronization process.
Has LDAP Group Mapping	Links LDAP Group Mapping records to Security Group records. This link determines who should belong to a given Security Group. Any user who belongs to the Active Directory group identified in a given LDAP Group Mapping record should belong to the Security Group to which it is linked. During the synchronization process, Meridium APM Security Users will be added to and removed from Meridium APM Security Groups based upon this link.
User Assignment	Links Security Group records to Security User records. The User Assignment relationship family is part of the standard Meridium APM security model. When you add a Security User to a Security Group via the Configuration Manager application, Meridium APM links the Security User record to the Security Group family via the User Assignment relationship. Similarly, when Security Users are added to Security Groups via the LDAP synchronization process, the two records are linked via the User Assignment relationship.
Interface Log	Stores information collected during the synchronization process. Records are created in this family only when synchronization logging is enabled.

Managing Users When LDAP Integration is Enabled

The LDAP integration feature is intended to simplify the Meridium APM user management process by allowing you to manage Meridium APM users via your existing, primary user management system: Active Directory. User information may change periodically in Microsoft Active Directory. For example, a user's address, phone number, or job title might change over time. One advantage of configuring LDAP integration is the ability to synchronize Meridium APM Security User records with the information in Active Directory so that changes made in Active Directory will be reflected in Meridium APM. LDAP integration is designed to ensure that these two systems (Meridium APM and Active Directory) are synchronized, *if you follow the recommended workflow for managing users*, which is:

1. Create users in Active Directory.

2. Assign users to Active Directory groups that are mapped to Meridium APM Security Groups.

3. Perform the synchronization process to create new users in Meridium APM. Any users added to Active Directory since the last synchronization will be created as Security Users in Meridium APM, provided that they are assigned to the Active Directory groups that are mapped to Meridium APM Security Groups.

4. Continue managing and modifying the users, as needed, *in Active Directory*. Changes made to the user accounts in Active Directory will be applied to the Security User records in Meridium APM when the synchronization process is run. *You should not make changes to the Security User records in the Meridium APM system.* Any changes that you make in the Meridium APM system will be overwritten when users are modified in Active Directory and the synchronization process runs.

> 🛈 **Note:** When you configure LDAP integration, you will create LDAP Mapping records to define the specific Active Directory user properties that correspond to Meridium APM Security User properties and will be updated during the synchronization process. Properties that are not mapped between the two systems will not be updated in Meridium APM automatically. You can manage these properties manually in Meridium APM.

A Note About User Status

When the LDAP synchronization process runs, a Meridium APM Security User's status (i.e., the value in the Status field of the Security User record) will be updated based upon various conditions in Active Directory. Meridium APM Security Users will be set to *inactive* when...

- Their Active Directory account is inactive.
- Their password has expired.
- They have been locked out of Active Directory (e.g., tried logging on too many times using the incorrect password).

The Meridium APM Security User will be *reactivated* automatically after these conditions are resolved in Active Directory and the synchronization process runs again. An LDAP Mapping record is not required for this behavior to occur. It will happen automatically regardless of the LDAP Mapping records that exist.

About the LDAP Synchronization Process

Synchronization will occur whenever the LDAP synchronization scheduled item is run. To facilitate automatic synchronization that will occur on a regular basis, you will need to create a scheduled item and set it up to run according to a schedule of your choice. The scheduled item will gather updated information from Microsoft Active Directory and then bring it into Meridium APM to update the corresponding Security User records.

To ensure that your Meridium APM system is in sync with your Active Directory system, we recommend that schedule the synchronization process to run on a frequent basis (e.g., every hour or more). A Meridium APM Security User's properties and status will always reflect the information that existed in Active Directory the last time the synchronization process was run. Running the synchronization process on an infrequent basis (e.g., weekly) will increase the likelihood that the information in Meridium APM will not match the information in Active Directory.

The synchronization process will pull into Meridium APM only the changes (i.e., new users and updated information) that have been made in Active Directory since the last synchronization ran, based upon the Last Execution date in the scheduled item. Because only *changes* are brought over to Meridium APM, the more often you run the synchronization process, the faster it will be (i.e., the fewer the changes, the faster the process). If you need to do a *full* update in Meridium APM, you will need to delete and recreate the scheduled item to clear out the Last Execution date. Performing a full synchronization will take longer than performing an *update* synchronization.

What Happens During Synchronization?

For synchronization to work properly, you will need to create various records and links in the Meridium APM Framework application. After the necessary records and links exist, when a synchronization operation is performed:

- The Meridium APM system will retrieve the information for the Active Directory users associated with the Active Directory domains and groups that have been defined in Meridium APM and will update the corresponding Security User and Human Resource records with any updated information. Fields in Meridium APM will be updated with the information in Active Directory using LDAP Mapping records.
- If the Meridium APM system finds a user in Active Directory who is a member of any of the Active Directory groups that have been identified through LDAP Group Mapping records but does not have a corresponding Security User record in Meridium APM:
  - A Security User and Human Resource record will be created in the Meridium APM database for the user.
  - The Security User record will be linked to the Domain record that identifies the Active Directory domain in which the user exists.
  - The Security User will be added to the Meridium APM Security Group to which that LDAP Group Mapping record is linked.

> **Note:** Because all new Security Users are automatically assigned to the Everyone Security Group, if the LDAP Group Mapping record is linked to a Meridium APM Security Group *other than* the Everyone group, the Security User will be added to that Security Group and the Everyone Security Group.

- If the Meridium APM system finds a user in Active Directory who is a member of any of the Active Directory groups that have been identified through LDAP Group Mapping records who already has a corresponding Security User in Meridium APM but does not belong to the Meridium APM Security Group to which that LDAP Group Mapping record is linked, the Security User will be added to the Meridium APM Security Group.

- If a user is removed from an Active Directory group, the corresponding Security User will be *removed* from the Meridium APM Security Group to which the LDAP Group Mapping record is linked.

A Note About Synchronization and Authentication

Meridium APM Security Users are authenticated at log-in. In addition to validating a user's status (Active or Inactive), at log-in, the Meridium APM system initializes all the user's information, Security Group assignments, and permissions. If any of that information changes while the Security User is logged in to the Meridium APM system, those changes will not be reflected immediately. The changes will not take effect until the user logs out of Meridium APM and then logs back in. This behavior applies to changes made manually via the Configuration Manager and changes made automatically via the LDAP synchronization process. In other words, regardless of when or how often the LDAP synchronization process runs, changes made to a user's account will not be applied until the next time a user logs in to the Meridium APM system.

Synchronization

When you configure LDAP integration, you have the option of enabling synchronization logging. When logging is enabled, an Interface Log record will be created each time the synchronization process is run to store information about that synchronization operation. The level of information that is stored in these records will depend upon the logging level that you select when you enable synchronization logging.

If errors are encountered during the synchronization process, that condition will be indicated in the Status field of the Interface Log record. After a user reviews the errors and resolves them, the value in the Status field can be changed to *Completed with Warnings (Cleared)* or *Completed with Errors (Cleared)*, as appropriate.

> **Note:** The Interface Log family is shared by several Meridium APM features and stores records that contain information about the status of other, non-LDAP interface operations. Interface Log records that were created by an LDAP synchronization process will always contain the value *LDAP Synchronization* in the Type field to make them easily distinguishable from Interface Log records created by other processes.

Overview of LDAP Configuration Tasks

To configure LDAP integration, you must complete the following steps. The records and links described in these steps are illustrated in the LDAP Integration Data Model.

Task	Reason
Enable LDAP Integration and logging.	LDAP integration will not be available until it has been enabled. If you enable LDAP integration, you will also want to determine the level of logging that you want to use. Logging is optional. If you do not enable logging, Interface Log records will not be created.
Configure LDAP integration to exlude domain names.	By default, when Security User records are created through LDAP synchronization, the value in the User ID field is set to *<name>@<-domain>*. If you want user IDs to contain only the name and *not* the domain, you can configure this setting manually.
Create a Domain record in Meridium APM for each Active Directory domain that contains users who also have Meridium APM user accounts.	Domain records store identifying information about the Microsoft Active Directory domains that exist in your organization.
Create LDAP Group Mapping records to identify the Active Directory groups that you want to use for synchronization.	LDAP Group Mapping records define the Active Directory groups that will be included in LDAP integration. Creating a Domain record alone is not sufficient for implementing LDAP integration. Only the users that exist within a define domain and belong to an Active Directory group defined by an LDAP Group Mapping record will be included in LDAP integration. Even if you want to retrieve users from ONE group, you must create an LDAP Group Mapping record to identify that group.
Link each LDAP Group Mapping record to a Domain record.	The link between an LDAP Group Mapping record and a Domain record identifies the group as belonging to the domain.

Task	Reason
Link each LDAP Group Mapping record to the Meridium APM Security Group record that identifies the corresponding Security Group in Meridium APM.	Users belonging to the Active Directory group defined in the LDAP Group Mapping record will be assigned to the Meridium APM Security Group defined by the record to which it is linked. For LDAP integration to work properly, at least one LDAP Group Mapping record must exist and be linked to a Security Group record. If this link does not exist, synchronization will not occur.
Review LDAP Mapping records.	These records specify how fields in Microsoft Active Directory should be mapped to fields in Meridium APM Security User records and Human Resource records. These records must exist for synchronization to work properly. A set of records exists in the baseline Meridium APM database, but you will need to review them to make sure that the mappings are appropriate for your implementation.
Link LDAP Mapping records to a Domain record.	The link between an LDAP Mapping record and a Domain record specifies that *that* mapping applies to *that* domain. In other words, if you want the user properties in an LDAP Mapping record to be updated for users that exist in a domain defined in a Domain record, you would link them together.
Create a scheduled item to periodically update Meridium APM with user information from Microsoft Active directory.	The scheduled item performs the LDAP synchronization process.

Enabling LDAP Integration and Logging

The first step in configuring LDAP integration is to enable the LDAP-based automatic login feature via a setting in the Configuration Manager application. Enabling LDAP integration will *disable* the default, non-LDAP automatic login feature, through which users are authenticated using their Windows user names and logged in automatically if a matching Meridium APM User ID exists. When you enable this setting in the Configuration Manager, you also have the option of enabling logging for the LDAP synchronization process. The synchronization process itself is enabled by completing other steps in the configuration process.

To enable LDAP automatic login and synchronization logging:

1. In the Configuration Manager, on the main menu, click **Tools**, point to **Global Preferences**, and then click **LDAP Integration**.

    The **LDAP Integration** dialog box appears.
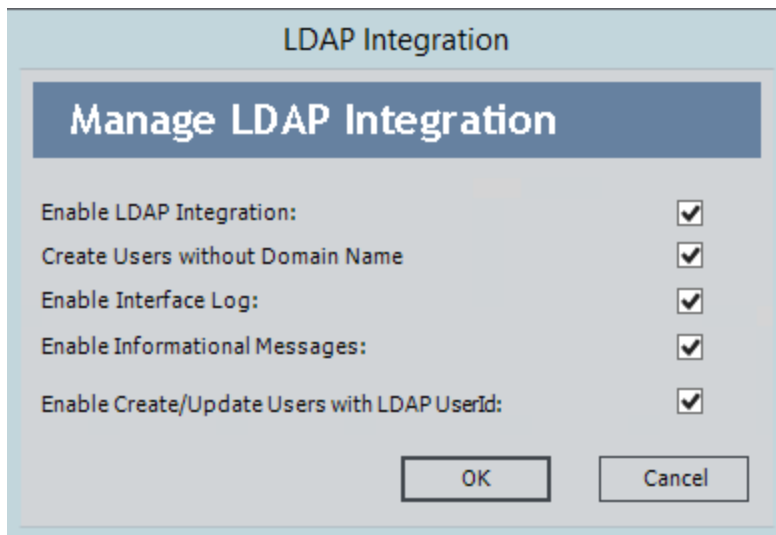
2. Select the **Enable LDAP Integration** check box.

    > ⓘ **Note:** If the **Enable LDAP Integration** check box is already selected, it means that LDAP Integration is already enabled. You can clear this check box to disable LDAP integration if it is already enabled.

3. If you want to enable synchronization logging, select the **Enable Interface Log** check box.

    > ⓘ **Note:** If this check box is not selected, Interface Log records will not be created with the synchronization process is performed.

4. If you want to enable detailed logging (i.e., debugging messages), select the **Enable Informational Messages** check box. This check box can be selected only if the

Enable Interface Log check box is also selected.

> ⓘ **Note:** This option controls the *level* of logging that will be performed. Interface Log records will be created as long as the **Enable Interface Log** check box is selected but will contain more detail in the Log Text field if the **Enable Informational Messages** check box is also selected.

5. If you want LDAP synchronization to occur based on the LDAP User ID instead of the Meridium User ID, select the **Enable Create/Update Users with LDAP UserId** check box.

> ⓘ **Note:** In certain situations, including when the Meridium User ID differs from the LDAP User ID, it may be preferable to have LDAP synchronization occur based on the LDAP User ID instead of the Meridium User ID.

6. Click **OK**.

Your selections are saved.

Configuring LDAP Integration to Exclude Domains from User Names

By default, when Security User records are created through LDAP synchronization, the value in the User ID field is set to *<name>@<domain>*. If you want user IDs to contain only the name and *not* the domain, you can configure this setting manually.

To configure LDAP integration to exclude domains from user IDs:

1. In the Configuration Manager, on the main menu, click **Tools**, point to **Global Preferences**, and then click **LDAP Integration**.

   The **LDAP Integration** dialog box appears.
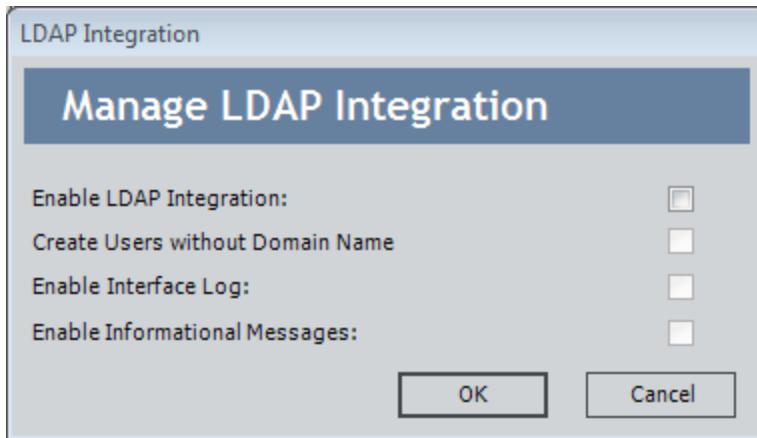
2. If it is not already selected, select the **Enable LDAP Integration** check box.

3. Select the **Create Users without Domain Name** check box.

4. Click **OK**.

   The setting is saved. Security User records will now be created automatically with the user ID format *<name>*.

Creating Domain Records

Domain records store identifying information about the Microsoft Active Directory domains that exist in your organization. For LDAP integration to work properly, at least one Domain record must exist to identify the Active Directory domain that contains user accounts that you want to synchronize with Meridium APM. You can create as many Domain records as needed to identify ALL the Domains from which you want to retrieve user information. Each Domain record must be linked to LDAP Mapping records and at least one LDAP Group Mapping record.

The baseline Meridium APM product contains a Domain record that you can use as the basis for creating the one required Domain record.

- If you have only one Active Directory domain, you can simply modify the baseline Domain record.

- If you have multiple Active Directory domains, you can modify the baseline Domain record and create new records to identify your additional domains. When you create a new Domain record, the default values will match those of the baseline Domain record to provide a guideline for specifying values in the new record.

Note that the baseline Domain record:

- Is already linked to the baseline LDAP Group Mapping record. If you are using *both* of these baseline records, you can simply accept the baseline link between them.

- Is not linked to the baseline LDAP Mapping records. You will need to configure these links manually even if you are using the baseline Domain record.

To create a Domain record:

1. In the Meridium APM Framework, on the Meridium APM Framework toolbar, click the **New** button.

   The **Select Family** dialog box appears.

2. In the list of families, select **Domain**.

3. Click **OK**.

   A new Domain record appears in the Record Manager.
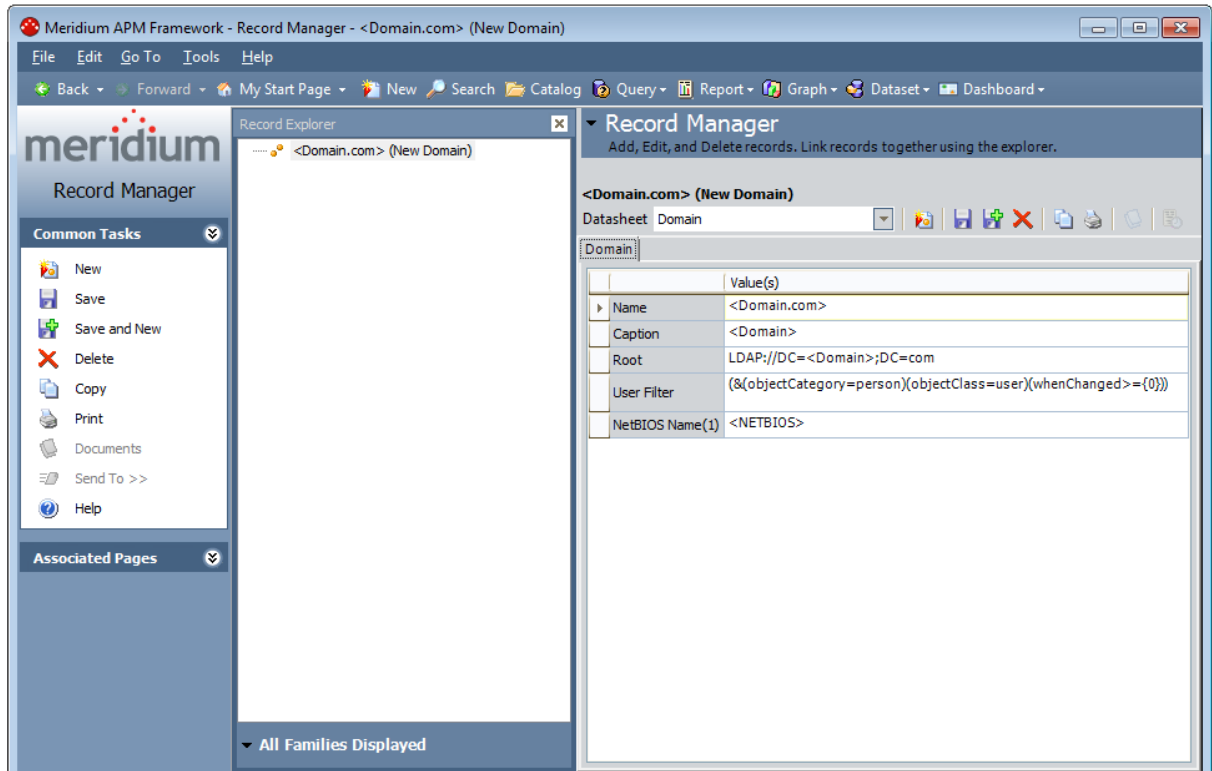
4. Define the appropriate values for the new record.

5. When you are finished, on the **Common Tasks** menu, click the **Save** link.

   The Domain record is saved to the Meridium APM database.

About LDAP Group Mapping Records

LDAP Group Mapping records identify the groups in an Active Directory domain from which you want to retrieve users for LDAP integration. LDAP Group Mapping records and the links between them and other records serve two purposes:

- The link between an LDAP Group Mapping record and a Domain record indicates that you want to retrieve users from that Active Directory group within that domain. LDAP integration does not require that you retrieve ALL users from a given domain. If only some of the users within a domain will use Meridium APM, you can include those users in one or more Active Directory groups and limit LDAP integration to those groups by creating LDAP Group Mapping records to identify them. Note that:

  - If you want to retrieve users from more than one Active Directory group, you will need to create at least one LDAP Group Mapping record for each group.

  - All LDAP Group Mapping records must be linked to one Domain record.

  - If users belong to more than one Active Directory group identified by an LDAP Group Mapping record, the Meridium APM system will reconcile the overlap and ensure that only single Security User record exists for that user in the Meridium APM database.

- The link between an LDAP Group Mapping record and a Security Group record identifies the Meridium APM Security Group to which users in a given Active Directory group should be assigned. When the LDAP synchronization process occurs, the Meridium APM system will retrieve users belonging to the Active Directory group identified by each LDAP Group Mapping record and assign them to the Meridium APM Security Group identified by the Security Group record to which each LDAP Group Mapping record is linked.

  - Each LDAP Group Mapping record must be linked to a Security Group record in order for the Meridium APM system to retrieve users from that group. If you want to assign all Security Users to a *single* Security Group, you should link the LDAP Group Mapping record to the Everyone Security Group record. All users will then be assigned to the Everyone Security Group.

  - LDAP Group Mapping records can be linked to ONE Security Group record only. So, if you want to assign users to *multiple* Meridium APM Security Groups, you will need to create *duplicate* LDAP Group Mapping records (one for each Meridium APM Security Group assignment that you want to make), link all of them to the same Domain record, and then link each one to the desired Security Group record. Each Security Group record can be linked to multiple LDAP Group Mapping records.

> **Note:** Because all new users are automatically assigned to the Everyone Security Group, all users created through the synchronization process and assigned to a group *other than* the Everyone Security Group will *also* be assigned to the Everyone Security Group. You need to create a mapping for the Everyone Security Group *only* if that is the only Security Group to which users should be assigned.

The baseline Meridium APM database contains a default LDAP Group Mapping record with the Record ID *<Your LDAP User Group Name>* that is already linked to the [baseline Domain record](). If you have modified the baseline Domain record to serve as your Domain record, you can modify the value in the LDAP Group Name field of the baseline LDAP Group Mapping record to serve as one of your LDAP Group Mapping records. If you do so and you need to retrieve users from only one Active Directory group, you simply need to link this baseline LDAP Group Mapping record to the Everyone Security Group record to complete this portion of the configuration process.

Creating and Linking LDAP Group Mapping Records

The following steps outline the process for managing a single LDAP Group Mapping record. You will need to repeat this process for each LDAP Group Mapping record that you need to create, taking into account the guidelines provided for LDAP Group Mapping records.

To create an LDAP Group Mapping record and link it to a Domain and Security Group record:

1. In the Meridium APM Framework, on the Meridium APM Framework toolbar, click the **New** button.

   The **Select Family** dialog box appears.

2. In the list of families, select **LDAP Group Mapping**.

3. Click **OK**.

   A new LDAP Group Mapping record appears in the Record Manager.
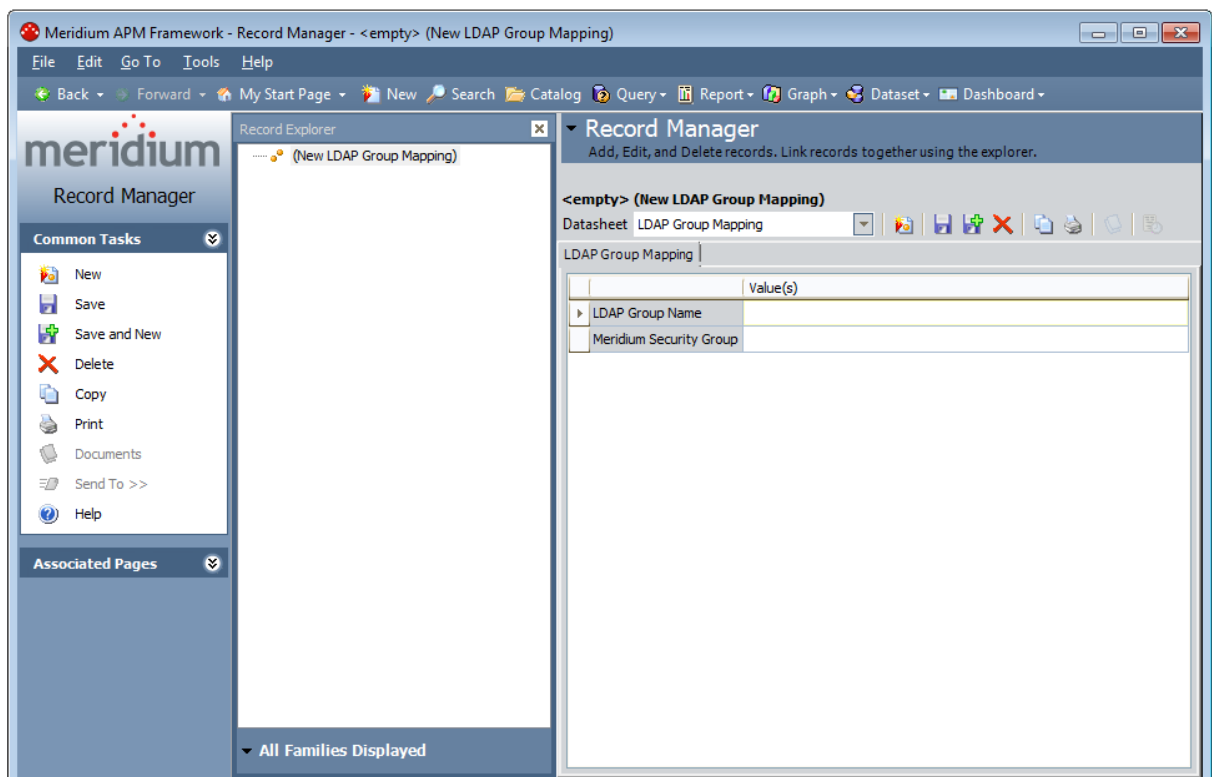
4. Specify values for the available fields, as desired.

5. On the **Common Tasks** menu, click the **Save** link.

   The LDAP Group Mapping record is saved to Meridium APM.

6. In the **Record Explorer** pane, right-click the **Domain** family node, and then click **Find an existing Domain record to link to <LDAP Group Mapping record>**.

   The **Find an existing Domain record to link to <LDAP Group Mapping record>** window appears.

   Note that, in the **Search In** list, **Domain** is selected by default.

7. Click the **Find Now** button.

   A list appears in the lower portion of the window, displaying all the Domain records.

8. In the results list, select the Domain record that you want to link to the LDAP Group Mapping record.

9. Click the **Link to Selected Record** button.

   The LDAP Group Mapping record is linked to the selected Domain record. The record hierarchy is updated to display an updated list of record links.

10. In the **Record Explorer** pane, right-click the **Security Group** family node, which appears directly below the LDAP Group Mapping record node, and then click **Find an existing Security Group record to link to <LDAP Group Mapping record>**.

    The **Find an existing Security Group record to link to <LDAP Group Mapping record>** window appears. Note that, in the **Search In** list, **Security Group** is selected by default.

11. Click the **Find Now** button.

    A list appears in the lower portion of the window, displaying all the Security Group records that are not currently linked to the selected LDAP Group Mapping record.

12. In the results list, select the Security Group record that you want to link to the LDAP Group Mapping record.

13. Click the **Link to Selected Record** button.

    The Security Group record that you selected in the search results are linked to the LDAP Group Mapping record that is currently open in the Record Manager. The record hierarchy is updated to display an updated list of record links.

About LDAP Mapping Records

LDAP Mapping records define how fields in Microsoft Active Directory user accounts cor-respond to fields in Meridium APM Security User records. The mappings that are defined in LDAP Mapping records are used to synchronize data between Active Directory and Meridium APM. The LDAP Mapping records tell Meridium APM what information should be retrieved from Active Directory and where it should be stored in Meridium APM. Each LDAP Mapping record contains the field *LDAP Field*, which defines the source Active Directory field, and a field *Meridium Field*, which defines the target Meridium APM field. Whenever synchronization occurs, data will be pulled from the source field (defined by the value in the LDAP Field field) and used to populate the value in the target field (defined by the Meridium Field field).

For example, consider the first row in the table below, which provides a list of the baseline LDAP Mapping records. The first row represents an LDAP Mapping record that contains the value *company* in the LDAP Field field and the value *MI_HR_COMPANY_CHR* in the Meridium Field field. Using this LDAP Mapping record, when synchronization occurs, the Meridium APM system will pull information from the *company* field in the appropriate Active Directory user accounts and use that data to populate the MI_HR_COMPANY_CHR field in the associated Security User records.

An LDAP Mapping record must exist for *each* Active Directory field that you want to map to a Meridium APM field. Meridium APM provides a set of baseline LDAP Mapping records that map standard Active Directory fields to fields in Meridium APM. If you want to map additional information to Meridium APM, you will need to create additional records. If you want to change the mappings that are defined through the baseline records, you can modify the records as needed. After you create the desired LDAP Map-ping records, you will need to link them to the appropriate Domain records.

> ⓘ **Note:** Whether you create your own LDAP Mapping records or use the baseline records, you will need to link them to the desired Domain records. The baseline LDAP Mapping records are not linked to the baseline Domain record by default.

The following table provides a list of mappings that are provided by the baseline LDAP Mapping records.

LDAP Field	Meridium Field
company	MI_HR_COMPANY_CHR
department	MI_HR_DEPT_CHR
givenName	MI_HR_FIRST_NAME_CHR
l	MI_HR_CITY_CHR
mail	MI_HR_EMAIL_TX
postalAddress	MI_HR_ADDR1_CHR

LDAP Field	Meridium Field
postalCode	MI_HR_POSTCODE_CHR
sn	MI_HR_LAST_NAME_CHR
st	MI_HR_STATE_CHR
telephoneNumber	MI_HR_PHONE1_CHR
title	MI_HR_JOB_TITLE_CHR

Linking LDAP Mapping Records to Domain Records

LDAP Mapping records define how fields in Microsoft Active Directory user accounts correspond to fields in Meridium APM Security User records. These fields may be specific to the Active Directory domain in which they are defined. Therefore, you will need to link each LDAP Mapping record to the appropriate Domain record.

You can create links between LDAP Mapping records and Domain records either by opening a Domain record and linking it to one or more LDAP Mapping records or by opening an LDAP Mapping record and linking it to a Domain record. In the following instructions, we provide details on opening a Domain record and linking it to one or more LDAP Mapping records.

To link an LDAP Mapping record to a Domain record:

1. In the Meridium APM Framework, on the Meridium APM Framework toolbar, click the **Search** button.

   The **Search** page appears.

2. If the **Simple Search** workspace is not already displayed by default, on the **Search Type** menu, click the **Simple search** link.

3. In the **Simple Search** workspace, in the **Search In** list, select **Domain**.

4. Click the **Find Now** button.

   The search results are displayed in the lower portion of the window.

5. In the list of results, in the **Record ID** column, click the hyperlinked Record ID of the Domain record that you want to link to one or more LDAP Mapping records.

   The selected record appears in the Record Manager.

6. In the record hierarchy, right-click the **LDAP Mapping** family node, which appears directly below the Domain record node, and then click **Find an existing LDAP Mapping record to link to <Domain record>**.

   The **Find an existing LDAP Mapping record to link to <Domain record>** window appears.

   Note that, in the **Search In** list, **LDAP Mapping** is selected by default.

7. Click the **Find Now** button.

   The search results are displayed in the lower portion of the window and include all the LDAP Mapping records that are not currently linked to the selected Domain record.

8. In the results list, select the LDAP Mapping record(s) that you want to link to the Domain record.

9. Click the **Link to Selected Record** button.

The LDAP Mapping records that you selected in the search results are linked to the Domain record that is currently open in the Record Manager. The record hierarchy is refreshed and displays an updated list of record links.

About Scheduling the Synchronization Process

For synchronization between Active Directory and Meridium APM to occur, you must create a scheduled item to manage the synchronization process. You can create this scheduled item via the Meridium APM Schedule Manager.

> ⓘ **Note:** This scheduled item will execute the synchronization operation only. For the operation to be successful, the necessary LDAP Mapping records must exist to map data from fields in Microsoft Active Directory to fields in the Meridium APM database.

To ensure that your Meridium APM system is in sync with your Active Directory system, we recommend that schedule the synchronization process to run on a *frequent* basis. A Meridium APM Security User's properties and status will always reflect the information that existed in Active Directory the last time the synchronization process was run. Running the synchronization process on an infrequent basis (e.g., weekly) will increase the likelihood that the information in Meridium APM will not match the information in Active Directory.

As with any scheduled item, when you create the scheduled item to manage the synchronization process, you will need to specify the Meridium APM Security User account that will be used for executing the scheduled item. You will need to specify a user who has permissions to perform the synchronization process.

Family-Level Privileges Needed to Run the Schedule Item

Wen you [create a scheduled item](#) to synchronize the information in Microsoft Active Directory with the Security User information in Meridium APM, you will need to specify a Meridium APM user account that will be used for executing the schedule item. The user that you specify must be a Super User.

Creating the Scheduled Item

To create the LDAP synchronization scheduled item in Schedule Manager:

1. Launch the Meridium APM Schedule Manager.

   The **Meridium APM Schedule Manager** window appears.
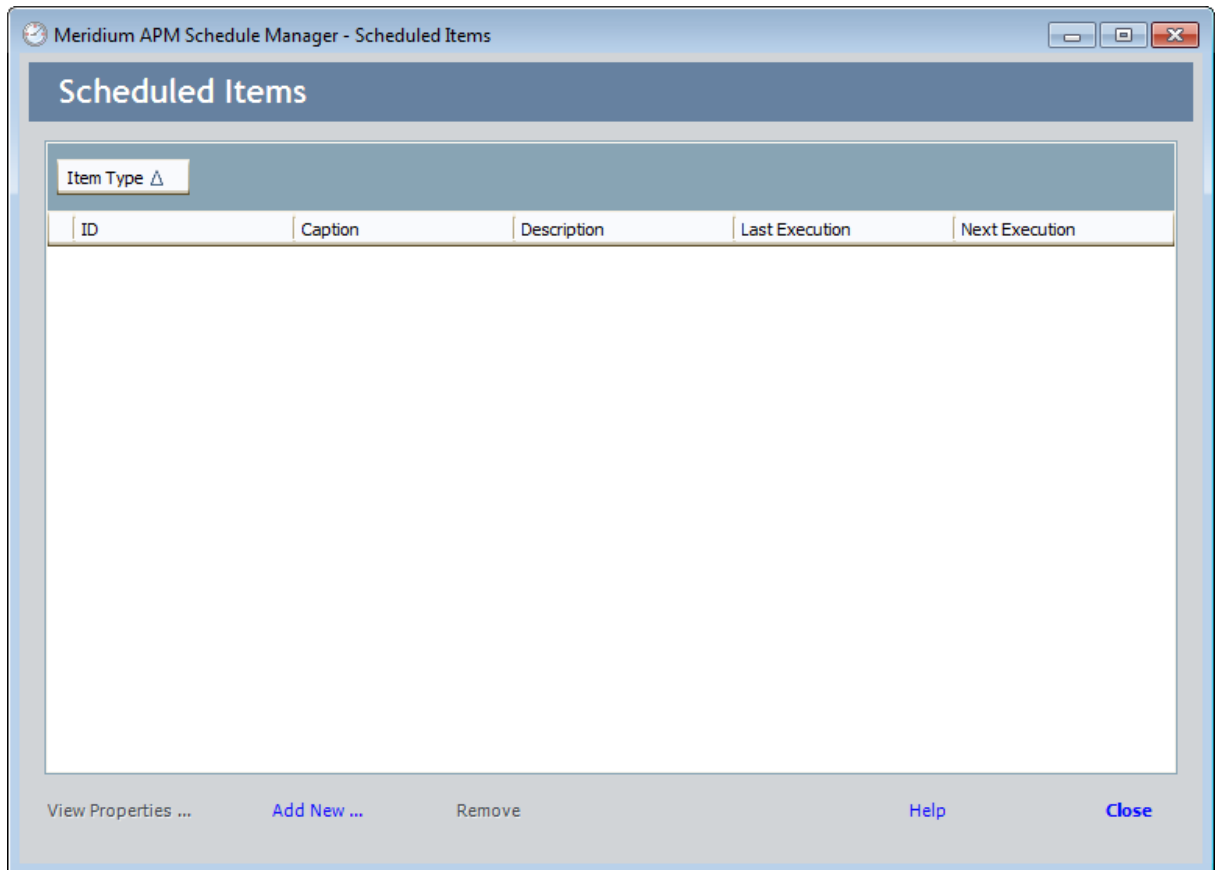
2. At the bottom of the **Meridium APM Scheduler - Scheduled Items** window, click the **Add New**link.

   The **Choose a Meridium Assembly** dialog box appears.
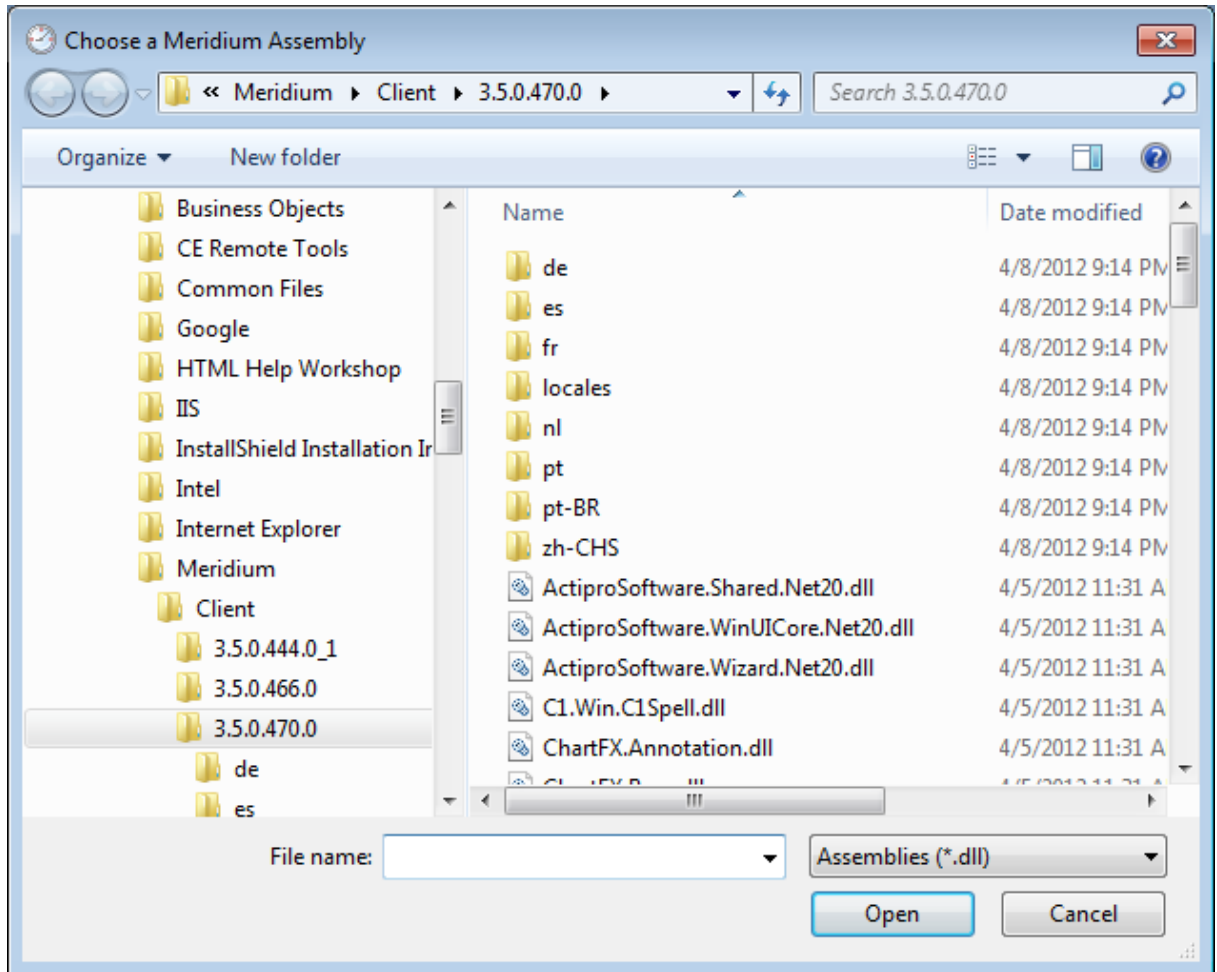
3. Open the file **Meridium.Registry.dll**.

    The **Choose an Object Type** dialog box appears.
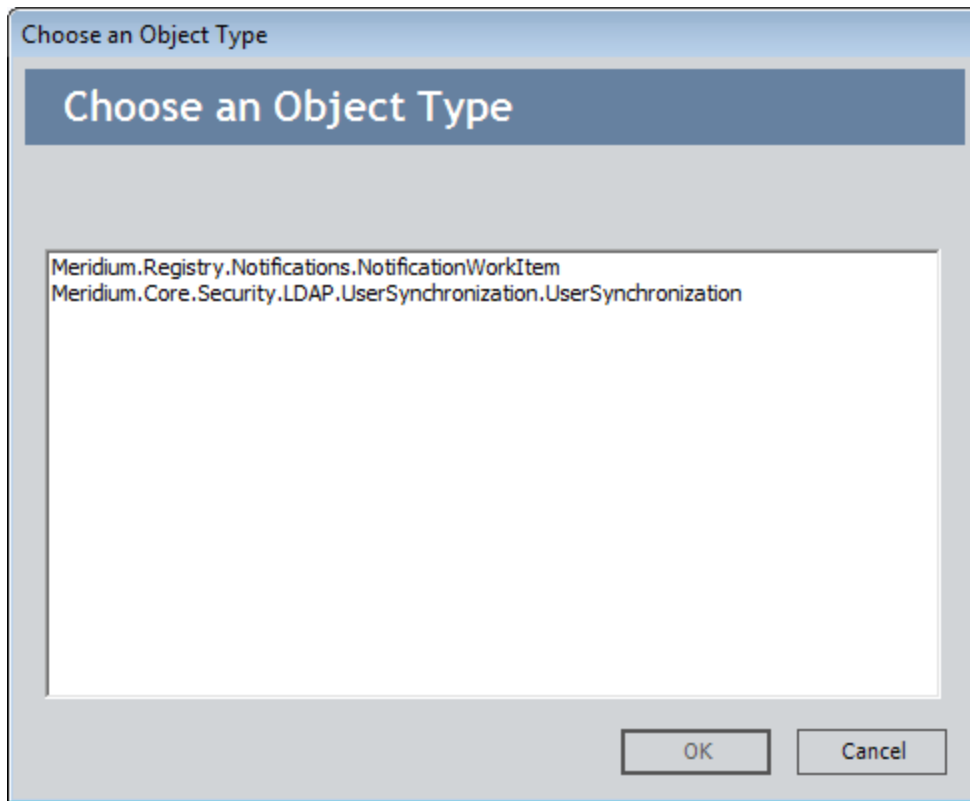
4. Select the file **Meridium.Core.Security.LDAP.UserSynchronization.UserSynchronization**, and click **OK**.

   The **Schedule User Synchronization** dialog box appears.
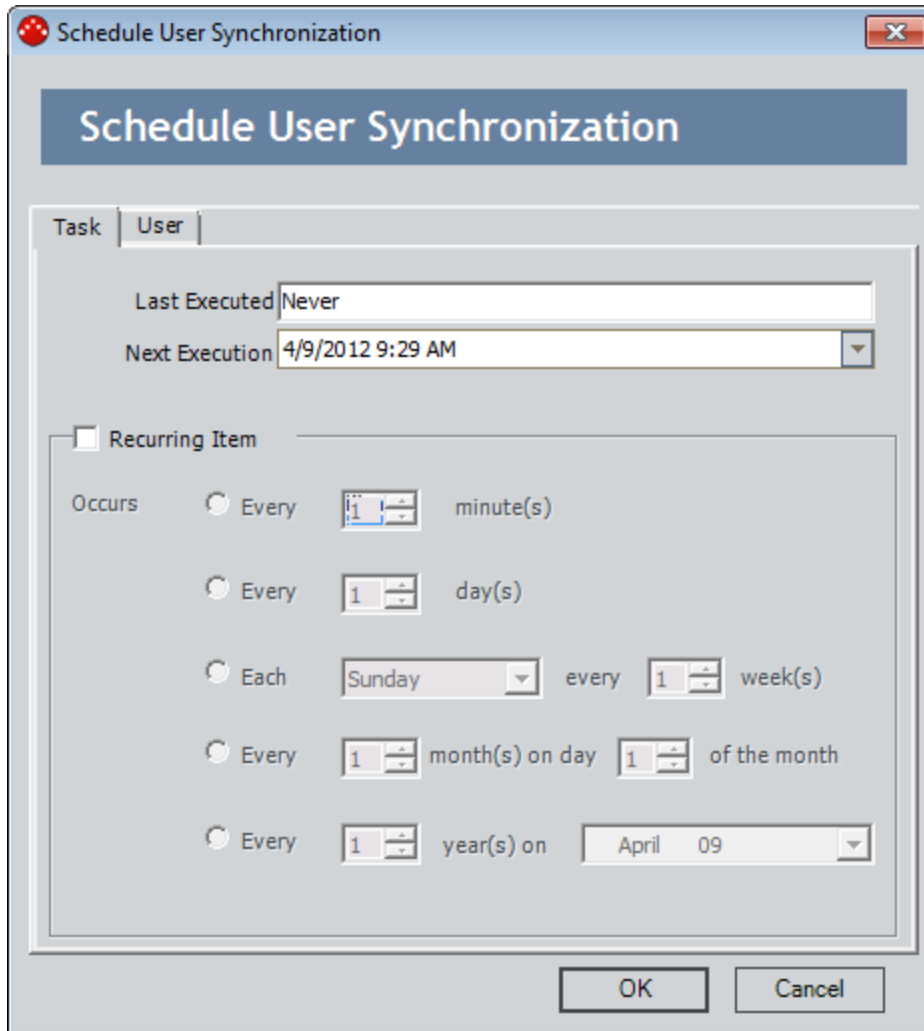
The **Last Executed** box contains the date on which the scheduled item was last executed. Since this is a new scheduled item that has never been executed, *Never* appears in this box.

5. Unless you want to start the synchronization process at some point in the future, accept the default selection in the **Next Execution** box (i.e., the current date and time). This means that the synchronization process will run as soon as the schedule item is saved.

> ⓘ **Hint:** You can delay execution by choosing a date in the **Next Execution** box that is in the future.

6. Select the **Recurring Item** check box.

The fields below the check box becomes enabled to allow you to specify the recurrence pattern.

7. To select the frequency with which the scheduled item will be executed, select the option button next to the desired pattern, and complete the fields as desired. The following table describes the schedules by which you can configure an item to be executed.

Frequency	Example	Default
Every x number of minutes.	Every thirty minutes.	Every minute.
Every x number of days.	Every two days.	Every day.
Every x day of every x number of weeks, where you define the day of the week.	Every Sunday of every two weeks.	Every Sunday of every week.
Every x month on day x of that month.	Every three months on day 10 of the month.	Every month on day 1 of the month.
Every x year on x date, where you define the date.	Every two years on March 15 of that year.	Every year on the current day of that year. For example, if the current date is January 9, the default value will be every year on January 9.

8. Click the **User** tab.

9. Type the Meridium APM User ID and password associated with the user account under which the synchronization process should run.

10. Click **OK**.

The schedule item is saved and will be executed the next time the Meridium APM Scheduling Service runs.

About Usage Metrics Tracking

By enabling usage metrics tracking, you can track the usage of users in your system. Usage metrics are stored in the MI_USAGE_METRICS system table. When a user logs in to Meridium APM, actions for which usage metrics tracking has been enabled will be stored for that session and saved in batch to the MI_USAGE_METRICS table when the user logs out of Meridium APM.

The following actions can be recorded in the MI_USAGE_METRICS table:

- Login.
- Logout.
- Session time.
- URL visit.

> ⓘ **Note:** Usage metrics are recorded only for activities performed via the Meridium APM Framework application. Usage metrics are not recorded for activities performed in the Meridium APM Administrative Applications.

For each action recorded in the table, the following columns of data are stored:

- **USME_KEY:** The key value assigned to the action to identify it in the usage metrics table.
- **USME_EVENT_TYPE_CD:** The type of event (login, logout, session time, or URL visit).
- **SEUS_KEY:** The key value associated with the Security User who performed the action.
- **USME_EVENT_DT:** The date and time the action was performed.
- **USME_EVENT_DESC_TX:** A description of the action. For URL visits, this column stores the URL.
- **USME_MEASR_NBR:** For session time entries, a numeric value that represents the session time.

After usage metrics tracking has been enabled and actions have been recorded in the database, you can use the Meridium APM Metrics functionality in the Meridium APM Framework application to view, format, and analyze the data.

Accessing Usage Metrics Tracking Settings

You can configure usage metrics tracking setting via the **Usage Metrics Tracking** dialog box in the Configuration Manager application.

To access the Usage Metrics Tracking dialog box:

- In the Configuration Manager, on the main menu, click **Tools**, point to **Global Preferences**, and then click **Usage Metrics Tracking**.

  The **Usage Metrics Tracking** dialog box appears.

From the **Usage Metrics Tracking** dialog box, you can [define settings for usage metrics tracking](#) and you can [clear the MI_USAGE_METRICS table](#).

Defining Settings for Usage Metrics Tracking

To define settings for usage metrics tracking:

1. In the Configuration Manager, on the Usage Metrics Tracking dialog box, select the check box for each action that you want to track. You can choose from the following actions:

   - **Login:** Adds an entry to the MI_USAGE_METRICS table each time a user logs in to the Meridium APM Framework application.

   - **Logout:** Adds an entry to the MI_USAGE_METRICS table each time a user logs out of the Meridium APM Framework application.

   - **Session Time:** For each login, records an entry for the amount of time that elapsed between the login and logout action.

   - **URL Visit:** Adds an entry to the MI_USAGE_METRICS table each time a user visits a Meridium APM URL.

   > **Note:** You can disable any currently enabled options by clearing the associated check box. Note that disabling the tracking of an action does not clear any previously saved actions from the MI_USAGE_METRICS table. It only disables the collection of future data.

2. When you have finished configuring the desired settings, click **OK**.

   The usage metrics tracking settings are saved.

Clearing the MI_USAGE_METRICS Table

The following instructions provide details on clearing the contents of the MI_USAGE_METRICS table. By using this procedure, you will permanently delete ALL the data stored in the MI_USAGE_METRICS table. This action cannot be undone.

⚠ **IMPORTANT:** You will not be asked for confirmation before the MI_USAGE_METRICS table is cleared.

To clear the data currently stored in the MI_USAGE_METRICS table:

- In the Configuration Manager, on the **Usage Metrics Tracking** dialog box, click the **Clear Usage Metrics** button.

  The usage metrics are deleted from the MI_USAGE_METRICS table.

Security Event Log

Meridium APM tracks security events for each user in your system. To view these security events, you can run the query *Security Event Log*, which is stored in the Catalog folder \\Public\Meridium\Modules\Core\Queries.

This query returns the following information for the Security Users in the Meridium APM database.

- When a user logs in successfully or fails to log in successfully
- When a user's password is changed
- When a user is created, activated, deactivated, inactivated, or modified

As you can see in the following image, the query returns this information in a table with headers that indicate the security event for each user.

Security Event Log : Select Query

Event Key	Event Name	User Id	Admin Id	Update Date
64252760878	UserCreated	chale	MIADMIN	12/2/2013 9:57:59 AM
64252760888	UserInactivated	chale	MIADMIN	12/2/2013 9:58:34 AM
64252760889	UserActivated	chale	MIADMIN	12/2/2013 9:58:50 AM
64252760890	UserModified	chale	MIADMIN	12/2/2013 9:58:56 AM
64252760891	PasswordChanged	chale	MIADMIN	12/2/2013 9:59:14 AM
64252760894	FailedLogin	chale	none	12/2/2013 10:03:13 AM
64252760895	SuccessfulLogin	chale	none	12/2/2013 10:03:17 AM

Domain Records

The following table provides a list and description of the fields that exist in Domain records. Most of these fields appear on the baseline Domain datasheet, with some exceptions (noted in the table). 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.

Field	Data Type	Description	Behavior and Usage
Caption	Character	A short description of the domain.	You can define this value manually to help distinguish this domain from any other domains that you define.
Group Filter	Character	N/A	This field is deprecated and no longer used by the LDAP integration feature.
Name	Character	The name of the domain.	You must define this value manually and should include the domain extension (e.g., domain.com). The value that you specify in this field will be used in conjunction with the @ symbol and Active Directory user names to create Meridium APM Security User IDs. For example, if a user's name isjsmithand the domain isMeridium.com, that users Meridium APM User ID would bej-smith@Meridium.com.
NetBIOS Name	Character	The NetBIOS name of the domain.	The NetBIOS name will be used by the Meridium APM system along with a user's LDAP user name to connect the user to that domain.
Root	Character	The starting point of the container in which Meridium APM will look for user objects in Active Directory.	The Meridium APM system will use this information to find user objects in Active Directory.
Single-User Filter	Character	N/A	This field is deprecated and no longer used by the LDAP integration feature.

Field	Data Type	Description	Behavior and Usage
Multi-User Filter	Character	The filter that should be used to locate users within the specified directory.	This filter will be used during the synchronization process to locate Active Directory users that belong to a specific group within the domain. You can accept the default value in this field. This field is labeled *User Filter* on the baseline Domain datasheet.

LDAP Mapping Records

The following table provides a list and description of the fields that exist in LDAP Mapping records and are available on the baseline LDAP Mapping 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.

Field	Data Type	Description	Behavior and Usage
LDAP Field	Character	The name of Microsoft Active Directory field that will serve as the source for the mapping.	You will need to define this value manually. This value must be defined manually. You can obtain a list of available Active Directory fields from Microsoft.
Meridium Field	Character	The field ID of the field in Meridium APM that will serve as the target field for the mapping.	You will need to define this value manually. The field can belong to any family, but you will probably want to specify a field that is defined in the Human Resource family or the Security User family. Be sure to specify the field ID, not the field caption.

LDAP Group Mapping Records

The following table provides a list and description of the fields that exist in LDAP Group Mapping records and are available on the baseline LDAP Group Mapping 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.

Field	Data Type	Description	Behavior and Usage
LDAP Group Name	Character	The Active Directory group from which you want to retrieve users for LDAP synchronization.	This field must be defined manually. The value must match exactly the name of an Active Directory group that exists in the domain defined by the Domain record to which the LDAP Group Mapping record is linked.
Meridium Security Group	Character	The name of the Meridium APM Security Group to which users in this Active Directory group will be assigned.	This field is optional and can be defined manually. The value in this field does not *determine* which Meridium APM Security Group users in this Active Directory group should be assigned; that is determined by the link between the LDAP Group Mapping record and the Security Group record. Because you must create an LDAP Group Mapping record for each Security Group mapping that you want to use, this field provides a visual indication of the mapping defined by the link.

Interface Log Records

The following table provides a list and description of the fields that exist in Interface Log records and are available on the baseline Interface Log 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.

Field	Data Type	Description	Behavior and Usage
Date Executed	Date	The date that the synchronization process was run and the time that the process finished running.	This field is disabled and populated automatically when the Interface Log record is created.
Log Text	Text	Information about the synchronization process, including any warnings or errors that occurred. The level of detail that appears in the Log Text field depends the level of logging that you selected when you enabled synchronization logging.	This field is disabled and populated automatically when the Interface Log record is created.
Number Of Records Created	Number	The number of Meridium APM Security User records that were created during the synchronization process.	This field is disabled and populated automatically when the Interface Log record is created.
Number Of Records Processed	Number	The number of Active Directory user records that were processed during the synchronization process.	This field is disabled and populated automatically when the Interface Log record is created.

Field	Data Type	Description	Behavior and Usage
Number Of Records Rejected	Number	The number of Active Directory users that were rejected during the synchronization process because an error condition was met.	This field is disabled and populated automatically when the Interface Log record is created.
Number Of Records Updated	Number	The number of Meridium APM Security User records that were updated during the synchronization process.	This field is disabled and populated automatically when the Interface Log record is created.
Parameters	Character	The Last Execution date from the scheduled item that ran the synchronization process. This is the time that the scheduled item was executed (i.e., when the synchronization process started).	This field is disabled and populated automatically when the Interface Log record is created.

Field	Data Type	Description	Behavior and Usage
Status	Character	The status of the synchronization process.	This field contains a list of values from the MI_CMMS_INF_ERROR_ CODES System Code Table. In the baseline Meridium APM database, the list contains the following System Codes: • **Completed:** The synchronization process completed successfully without warnings or errors. • **Completed with Warnings:** The synchronization process completed successfully, but with warnings, indicating that you might want to review the warnings to determine if any action is needed. • **Completed with Errors:** The synchronization process did not complete successfully. • **Completed with Warnings (Cleared):** The synchronization process completed successfully with warnings, and someone has reviewed the warnings. • **Completed with Errors (Cleared):** The synchronization process did not complete successfully, and someone has reviewed the errors. When the synchronization process is finished running, the value in the Status field will be set automatically to *Completed*, *Completed with Warnings*, or *Completed with Errors*. If the status indicates that warnings or errors have occurred, you can

Field	Data Type	Description	Behavior and Usage
			review the warnings or errors and then change the status to *Completed with Warnings (Cleared)* or *Completed with Errors (Cleared)*, as appropriate.
System ID	Character	This field is not used for Interface Log records created by the LDAP synchronization process.	This field is disabled and populated automatically when the Interface Log record is created.
Type	Character	The name of the process that was executed. For Interface Log records created by the LDAP synchronization process, this field will always contain the value *LDAP Synchronization*.	This field is disabled and populated automatically when the Interface Log record is created.

Security User Properties

Security User properties determine the specific information that is associated with a Security User account. You can configure properties for a new Security User or modify the properties of an existing Security User on the **Application User** dialog box. The following table provides a list and description of the Security User properties that you can define on the **User Information** tab of the **Application User** dialog box.

Security User Property	Description	Notes
User ID	A unique value that identifies the Security User within the system and be used for logging in to the Meridium APM applications. This field is required, must be unique, and has a limit of 50 characters.	• If LDAP integration has not been enabled, User IDs serve as the trigger for invoking the Meridium APM non-LDAP automatic login feature. When a user launches a Meridium APM application that supports automatic login, Meridium APM checks the user name that the user provided for logging in to the Windows operating system. If the default Meridium APM database contains a Security User with a User ID that matches that user's Windows user name, the user will be logged in to Meridium APM automatically under the associated Security User account, without having to specify a user ID and password. • If LDAP integration has been enabled, User IDs help identify users within the Meridium APM system but do not invoke automatic login. Instead, you will want to make sure that the value in the **LDAP User** field matches the user's Microsoft Active Directory user name. The **LDAP User** ID will be used in combination with the associated Domain record to determine whether or not the user will be logged in automatically.

Security User Property	Description	Notes
LDAP User	The Microsoft Active Directory user name for this user. This value is optional and needs to be supplied only if LDAP integration is enabled. The value specified in this field will be used to identify this user in an Active Directory domain.	N/A
Auto-populate LDAP User ID	A setting that determines how the **LDAP User** text box will behave. • When the **Auto-populate LDAP User ID** check box is selected, the **LDAP User** text box will be disabled and populated automatically with the value specified in the **User ID** text box. • When the **Auto-populate LDAP User ID** check box is cleared, the **LDAP User** text box will be enabled so that you can type any value of your choice.	When you create a new Security User record, the **Auto-populate LDAP User ID** check box will be selected by default, but you can clear the check box if you do not want the LDAP User ID to match the Meridium APM User ID. When you open an existing Security User record: • If the Meridium APM User ID and LDAP User ID *match*, the **Auto-populate LDAP User ID** check box will be selected by default. You can clear the check box if you want to specify an LDAP User ID that is different from the Meridium APM User ID. • If the Meridium APM User ID and LDAP User ID are *different*, the **Auto-populate LDAP User ID** check box will be cleared by default. If you select this check box, the existing LDAP User ID will be overwritten with the value that exists in the **User ID** text box.

Security User Property	Description	Notes
Password	The password for the Security User. Passwords are case sensitive and can contain any character. There is no minimum requirement for passwords (i.e., you can leave this field blank), but they cannot exceed 18 characters in length. If you type a password in the **Password** text box, when you move your cursor to a different field, the **Confirm Password** dialog box will appear, prompting you to verify the password by re-typing it.	• When you create a new user, you can leave the **Password** text box blank, in which case the user will be created without a password. Security Users who have no password will be able to log in to Meridium APM by supplying their User IDs only. After a user has been created without a password, when you access the **Application User** dialog box for that user, the **Blank Password** check box will be selected, and the **Password** text box will be disabled. If you want to define a password for a user who does not have one, you can clear the **Blank Password** check box and then type the desired password in the **Password** text box. • If you are using automatic login (either LDAP or non-LDAP automatic login) in your Meridium APM system, users will not be required to enter a user name or password when they log in to a Meridium APM application that supports automatic login. For automatic login, the Meridium APM password will be ignored. • Passwords are stored with scheduled items that are created via the Meridium APM Schedule Manager. If you change a user's password, any scheduled items associated with that Security User will be updated with the new password.

Security User Property	Description	Notes
Query Privilege	A setting that indicates the restrictions that exist for the Security User when they are working with queries. This value can be set to: • **Unrestricted:** The Security User can save new or modified queries without restriction. • **Restricted By Timeout Limit:** The Security User can save new and modified queries only if the query results are returned within a specified amount of time. • **Execute Only:** The Security User cannot create or modify queries. Instead, he or she can only view the result of existing queries. • **None (empty):** Has the same effect as setting the value to *Unrestricted*.	This list is populated using the Query Privileges (MI_QUERY_PRIV) System Code Table. If the **Super User** check box is selected, the value is cleared automatically, and the field becomes disabled.
Status	A setting that indicates the status of the Security User, either: • **A** (active) means that the Security User can use the system. -or- • **I** (inactive) means that the Security User cannot access the system or any of its data. Inactive users can be reset to Active if they are later needed again.	Security Users cannot be deleted. Users that are no longer needed in your system should be set to Inactive.

Security User Property	Description	Notes
Super User	Specifies whether or not the Security User is a Meridium APM Super User. Super Users have full access to ALL features in the Meridium APM application without requiring additional family-level permissions.	N/A
Timezone	The time zone associated with the Security User.	This field contains a list of time zones that correspond with the time zones that are displayed in your Windows Region and Language settings. The Meridium APM system uses the value that you select in this list in conjunction with the UTC? field property to determine how to convert the UTC time value and display the value to a user in their local time. For example, the UTC? field property is selected by default for the End Date field in the Production Plan family. As such, when a user whose time zone is (UTC-05:00) Eastern Time (US & Canada) enters an End Date of *12/31/2008 3:04PM* , this value is shown as *12/31/2008 12:04PM* to a user whose time zone is (UTC-08:00) Pacific Time (US & Canada).
Culture	A setting that determines which translated strings will be displayed to this Security User if you are using translations in Meridium APM. This value is optional.	If you do not select a value for this setting, the Meridium APM applications will be displayed in the default language, English. Throughout the Meridium APM documentation, we refer to the value in this field as the *Meridium APM Culture setting*.

Security User Property	Description	Notes
UOM Conversion Set	Indicates which UOM Conversion Set you want to associate with this Security User. The UOM Conversion Set that you select determines how values stored in one UOM will be converted for display in another UOM for this Security User. The **UOM Conversion Set** list displays all the UOM Conversion Sets that have been defined for this database. The **UOM Conversion Set** property is required.	By default, the **UOM Conversion Set** property is set to **Meridium APM** for new Security Users, indicating that they will use the default **Meridium APM** Conversion Set. If you have configured additional Conversion Sets, you can modify the default value. Otherwise, simply accept the default.
Last Name	The last name of the person for whom you are creating a user account.	This field is required and must be defined for all users.
Additional Personal Information	Additional information that further identifies the Security User within the system, including the user's address, phone number(s), and job information.	This information is optional.

When you are modifying an existing Security User that has been saved, you can click the **Edit Home Page** link at the bottom of the **User Information** tab to log in to the Meridium APM Framework application and modify the Security User's Home Page. Additionally, when you are modifying an existing Security User, you can click the **Groups** tab to view the Security Groups to which you can assign a Security User. All Security Users are assigned by default to the **Everyone** group.

System Code Tables Used By the Security User Family

The Security User family uses the following System Code Table:

- **Table ID:** MI_QUERY_PRIV
- **Table Description:** Query Privileges

This System Code Table is used to populate the **Query Privileges** list in Security User records. The Meridium APM system provides baseline System Codes with the following descriptions:

- Unrestricted
- Restricted By Timeout Limit
- Execute Only

> **Note:** We recommend that you *not* change the MI_QUERY_PRIV System Code Table or any of its System Codes. If you do, the Meridium APM system will not function as designed.

About the Global Number and Date Format

By default, throughout the Meridium APM product, numeric and date fields are displayed in the following ways:

- Numeric fields show ALL the decimal places that are stored in the field, with no rounding. For example, if a field contains the value 1.2345678912345, the value 1.2345678912345 will be shown wherever that field is displayed.

- Date fields are displayed using a combination of the Short Date and Long Time formats defined in your Windows Region and Language settings (i.e., both the short date and the long time are displayed for each date field). For example, if your Short Date format is set to mm/dd/yyyy and your Long Time format is set to hh:mm:ss:tt, date fields would be displayed as 1/1/2012 01:00:00 AM.

You can modify this default behavior in two ways:

- [You can define a global setting to control the display of ALL fields throughout the product](#).
    - For example, you might define a global format for numeric fields to specific that *only* two decimals places should be displayed.
    - For example, you might define a global date format that *excludes* time information since, for most date fields, the time is irrelevant.

- [You can define field-level Format rules to control the display of individual fields](#). Field-level format rules are specific to a given field and will override the default behavior and the global setting (if one is defined).
    - For example, if the global number format is configured to display two decimal places, for a specific numeric field that is intended to be more precise, you might define a Format rule to display *four* decimal places.
    - For example, if the global date format *excludes* time information, for a specific date field where time information is important (e.g., if it is important to capture the specific time at which an event occurred), you might define a Format rule to *include* time information.

The following approach is used to determine which display format to use:

- If a global setting is not defined, the default format will be used except for fields where field-level format rules have been defined. Those fields will be displayed according to their Format rules.

- If a global setting is defined, it will be used on all fields except fields for those where field-level format rules have been defined. Those fields will be displayed according to their Format rules.

> **Note:** The global setting is stored as a system preference but will be applied in all locations where format rules are applied (e.g., datasheets and queries running in Formatted mode).

Defining a Global Number and Date Format

Global formats for numeric and date fields can be defined independently of one another. You can define a global number format, a global date format, or both. Because you define these settings in the same location, however, the following instructions provide details on defining both.

To define a global format for numeric and date fields:

1. In the Configuration Manager, on the main menu, click **Tools**, point to **Global Preferences**, and then click **Global Number and Date Format**.

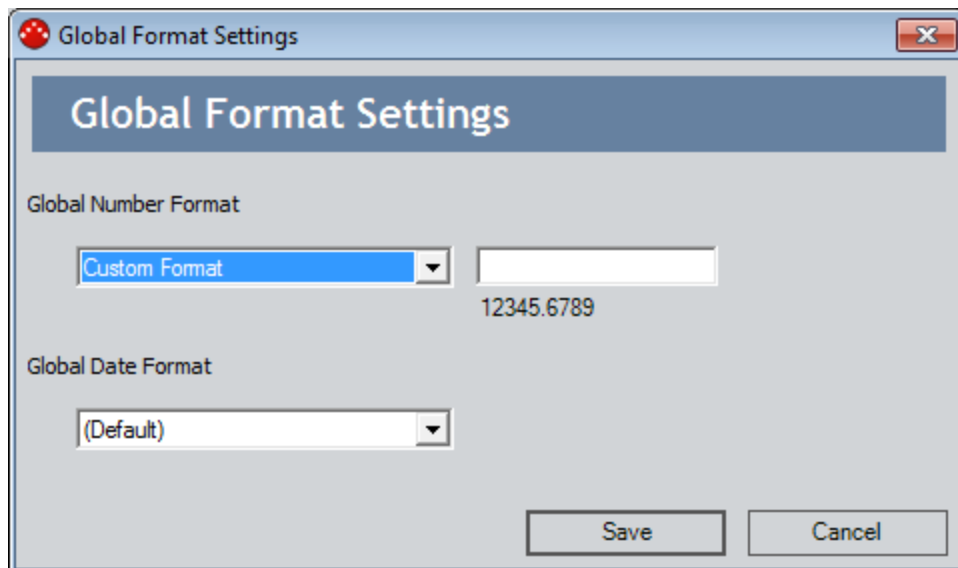   The **Global Format Settings** dialog box appears.
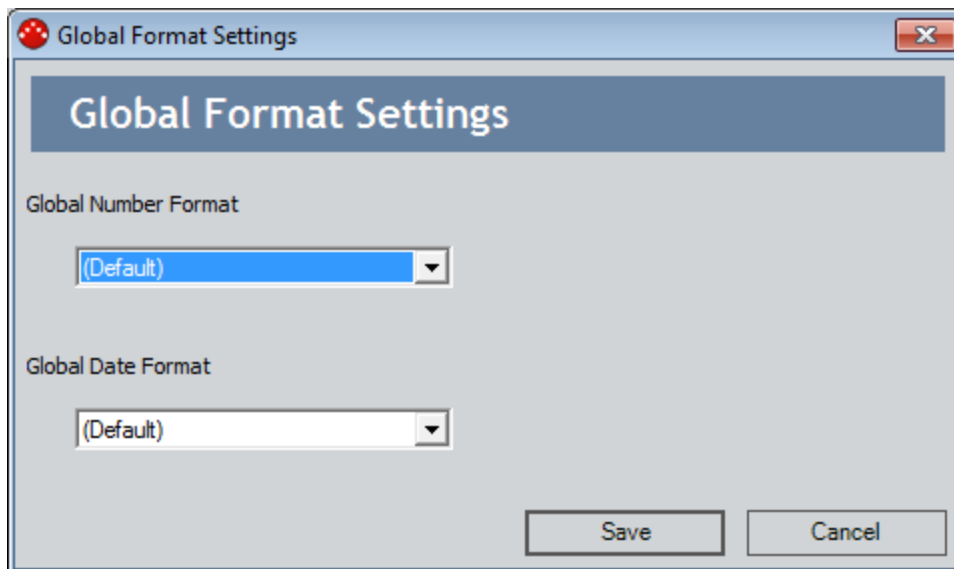
2. In the **Global Number Format** list, choose the option that corresponds to the type of format that you want to use for numeric fields. Click here for details on the available options.

   - **(Default):** Specifies that the Meridium APM system will display numeric fields using the default format (i.e., all decimal places that exist in the field). This option is selected by default. You can accept the default selection if you choose not to define a global number format. Additionally, you can choose this option to clear any global number format that is currently defined and revert to the default behavior.

   - **Fixed Precision:** Allows you to specify the number of decimal places that will appear for numeric fields throughout the Meridium APM product. When you choose this option, an additional box appears to the right of the **Format** list:
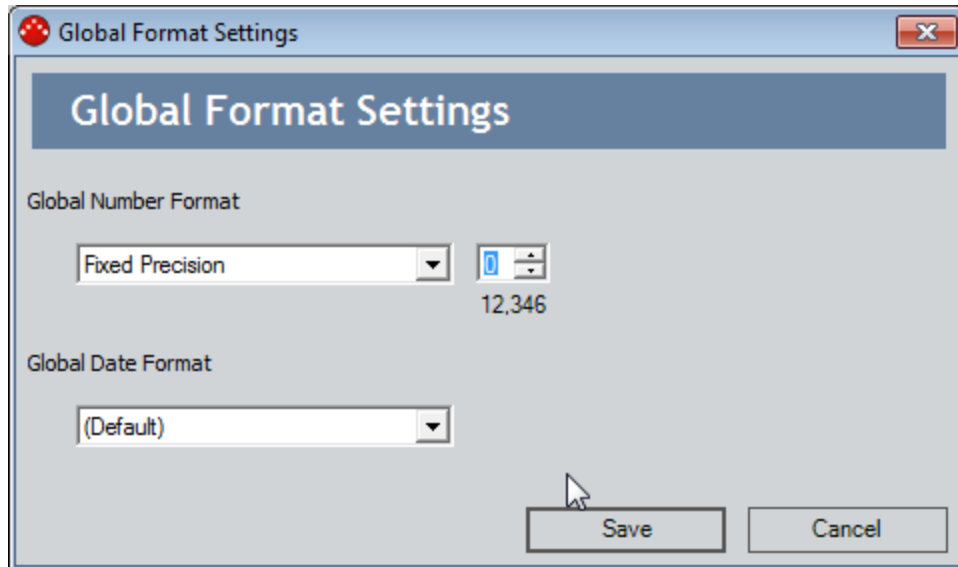
In this box, you can type or choose a number that indicates the number of decimal places that should be displayed. As you modify the value in this box, an example appears below it to indicate how numbers will be displayed.

- **Custom Format:** Allows you to define a custom number format. When you choose this option, an additional box appears to the right of the **Format** list:

With a custom format, you can define one or more characters to appear either to the right or left of numeric values. Additionally, you can define the number of digits that will appear to the right of the decimal by typing .0000, where each zero that you type indicates a decimal place that will be displayed. You can type as many zeros as you want. As you modify the value in

this box, an example appears below it to indicate how numbers will be displayed.

3. In the **Global Date Format** list, choose the option that corresponds to the type of format that you want to use for numeric fields. Click here for details on the available options.

- **(Default):** Specifies that the Meridium APM system will display date fields using the default format (i.e., the Short Date and Long Time defined in your Windows Region and Language settings). This option is selected by default. You can accept the default selection if you choose not to define a global date format. Additionally, you can choose this option to clear any global date format that is currently defined and revert to the default behavior.

- **Short Date:** Specifies that the short date format will be used and that time will be excluded. Example: 3/31/2005

- **Long Date:** Specifies that the long date format will be used and that time will be excluded. Example: Thursday, March 31, 2005

- **Short Time:** Specifies that the short time will be displayed and that dates will be excluded. Example: 4:27 PM

- **Long Time:** Specifies that the long time will be displayed and that dates will be excluded. Example: 4:27:00 PM

- **Date and Time Short:** Specifies that the long date and short time will be displayed. Example: Thursday, March 31, 2005, 4:27 PM

- **Date and Time Long:** Specifies that the long date and long time will be displayed. Example: Thursday, March 31, 2005, 4:27:00 PM

- **General Short Time:** Specifies that the short date and short time will be displayed. Example: 3/31/2005 4:27 PM

- **General Long Time:** Specifies that the short date and long time will be displayed. Example: 3/31/2005 4:27:00 PM

- **RFC 1123:** Specifies that the RFC 1123 date and time will be displayed. Example: Thu, 31 March 2005 16:27:00 GMT

- **ISO 8601:** Specifies that the ISO 8601 date and time will be displayed. Example: 2005-03-31T16:27:00

- **Month Day:** Specifies that only the month and day will be displayed and that the time will be excluded. Example: March 31

- **Year Month:** Specifies that only the month and year will be displayed and that the time will be excluded. Example: March, 2005.
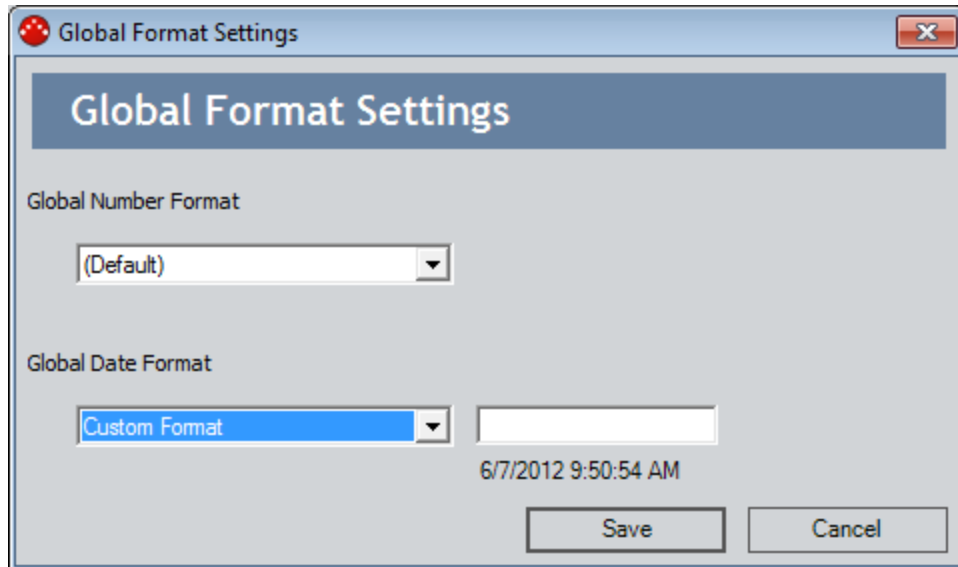
- **Custom Format:** Allows you to define a custom date format. When you choose this option, an additional box appears to the right of the **Global Date Format** list:

The format that you specify in the text box determines how dates will be displayed. An example appears below the text box as you modify the value in it. Use the following characters to specify each digit in the format.

- **M:** Month.

- **d:** Day.

- **Y:** Year.

- **H:** Hour.

- **s:** Second.

4. When you are finished defining the desired formats, click the **Save** button.

Your changes are saved.

Introduction to URLs

In the most common terms, a **Uniform Resource Locator (URL)** is an address that identifies a document or file on the World Wide Web. You can use URLs to construct links that also provide access to specific files using *any* application that supports the use of URLs, such as Meridium APM.

You are probably already familiar with how URLs are used on the World Wide Web. Whenever you click a link on the Internet, your web browser takes you to the URL that is defined for that link. Whenever you access a webpage, the URL for that page appears in the **Address** field of your web browser. At one time or another, you have probably copied a URL from the **Address** field and sent it to someone via email or used it to create a link on another webpage so that other users could access the associated resource.

In the Meridium APM Framework application, you can use URLs in a way that is similar to how they are used on the World Wide Web. Just like every location on the World Wide Web is URL-addressable, most locations in Meridium APM are also URL-addressable. This means that you can use a URL to view any page, open any item, and access any feature in Meridium APM that supports the use of URLs. In fact, in many cases, when you click a button, a menu item, or a hyperlink in the Meridium APM Framework application, you are actually executing a URL that is stored in the system. The URL tells the system where to take you or which function to execute.

The URL functionality in Meridium APM gives you a high degree of flexibility in configuring and customizing your system. After you learn how to configure URLs in Meridium APM, you can:

- [Add hyperlinks to Personal and Group Home Pages](#) to provide quick access to commonly used features and functions.
- Add hyperlinks to queriesso that your query results contain links that provide more detailed information about each specific item returned in the results set.
- Add customized hyperlinks to graphs so that you can view additional information about the results depicted in a chart.
- Create links on the **Associated Pages** menufor families.

> ℹ **Note:** In Meridium APM, you can create links using both internal URLs (i.e., URLs for Meridium APM modules) and external URLs (i.e., URLs for external documents, such as those that reside on the World Wide Web).

In this documentation, our discussion of Meridium APM URLs is limited primarily to how URLs are used in the Meridium APM Framework application. We provide only a limited discussion of URLs as they relate to Meridium APM Web Framework. Where it is not stated explicitly, you can assume that our discussion of URLs and their behavior apply to the Meridium APM Framework application.

The Basic URL Syntax

In the most basic terms, every URL consists of two main parts:

- The *scheme*, which specifies the method that will be used to open the document or file identified by the URL.
- The *scheme-specific part*, which identifies exactly what to open using the specified scheme.

The scheme-specific part is always separated from the scheme by a colon. For example, consider the following URL, which opens the Meridium, Inc. website:

http://www.meridium.com

In this example, the *scheme* is **http**, and the *scheme-specific part* of the URL is **//www.meridium.com**, where the two are separated by a colon. A link constructed using this URL will launch the Meridium, Inc. website in a web browser using the Hypertext Transfer Protocol (HTTP), which is the underlying protocol used by the World Wide Web. Other schemes you might already be familiar with using include **https**, **file**, and **mailto**.

Meridium APM has its own scheme, which is used for accessing Meridium APM modules via Meridium APM URLs. The scheme used in constructing Meridium APM URLs is **meridium**. For example, consider the following URL for the **Inspection Management Start Page** in Meridium APM:

meridium://Inspection

In this URL, the *scheme* is **meridium**, and the *scheme-specific part* is **//Inspection**. A link constructed from this URL will launch the **Inspection Management Start Page** in Meridium APM.

Just like every document on the World Wide Web has its own unique URL, each feature in the Meridium APM Framework application has its own address. In this way, you can use Meridium APM URLs to construct links that access specific features in Meridium APM. The purpose of this documentation is to provide details on constructing addresses that will access the Meridium APM modules.

> 🛈 **Note:** Throughout this documentation, we use certain capitalization conventions within URLs to help make our examples clear and legible. Note, however, that most URLs are case *in*sensitive, meaning that as long as a URL contains the correct syntax, capitalization does not matter. *There are some exceptions to this general rule.* These exceptions are noted throughout this documentation, where appropriate.

Extending the Basic URL Syntax

The scheme-specific part of a URL will contain information using a syntax that can be interpreted by the specified scheme. For example, the scheme-specific part for URLs that use the scheme **http** is defined as:

//<user>:<password>@<host>:<port>/<path>?<searchpart>

Depending upon which specific information the URL is designed to access, certain components of the scheme-specific part may be omitted. For example, in general, when you construct a URL to access a particular webpage, you omit the **<user>**, **<password>**, and **<port>** components of the syntax. You can omit portions of scheme-specific part when they are not required for the file you are trying to access or when you want to use certain default values that are defined for the web server or host.

For example, consider the following URL where the **<user>**, **<password>**, and **<port>** components have been omitted:

http://www.meridium.com/news_events/news/press_releases.asp?press_release_ID=47.

In this URL:

- The **<host>** is **www.meridium.com**.
- The **<path>** is **news_events/news/press_releases.asp**.
- The **<searchpart>** is **press_release_ID=47**.

Similarly, the scheme-specific part of URLs that use the **meridium** scheme is defined as:

//<user>:<password>@<data source>/<path>?<parameters>

Because most Meridium APM URLs will be used within Meridium APM itself (e.g., in queries and on Home Pages), you will usually omit the **<user ID>**, **<password>**, and **<data source>** when you construct Meridium APM URLs. When users log in to Meridium APM, they specify a user name, password, and data source. When a user clicks a link constructed from a URL where the **<user ID>**, **<password>**, and **<data source>** have been omitted, The Meridium APM sytem will use the user name, password, and data source that was supplied at login.

For example, if the following URL existed on your Home Page, when you clicked it, the system would use your current connection information to open the specified RCM analysis: **meridium://RCM/Explorer?EntityKey=1234567&FamilyKey=2345678**.

In this example:

- The **<path>** portion is **RCM/Explorer**
- The **<parameters>** portion is **EntityKey=1234567&FamilyKey=2345678**

> **Note:** For URLs that will be accessed from outside the Meridium APM Framework application, you can omit the **<user ID>**, **<password>**, and **<data source>**, which will

cause the user to be prompted to enter the necessary login information before the associated page is displayed. Alternatively, you can include the **<user ID>**, **<password>**, and **<data source>**, which will cause the user to be logged in automatically using the supplied information. Throughout the rest of this documentation, we omit the **user**, **password**, and **data source** from our examples and descriptions of Meridium APM URLs and limit our discussion to paths and parameters.

This documentation provides details on the URL paths that are used in Meridium APM, descriptions of the parameters that can be used with each path, and examples that combine paths with parameters to access specific features. This documentation, however, does not provide a specific example for accessing every feature in Meridium APM. The URLs that you construct will vary from our examples, depending upon your unique implementation. By understanding some general characteristics of URL syntax, however, you can customize and extend the generic information and examples that we provide to meet the needs of your organization within the parameters of your system.

The URL Path

The path of a URL specifies the main area of Meridium APM that will be accessed by the URL. For example, the URL **meridium://Registry/ReportViewer**, where the path is **Registry/ReportViewer**, accesses the Meridium APM Report Viewer. The path generally represents a particular feature or page in the Meridium APM Framework application. Each path corresponds to a registered URL, which is defined via the Meridium APM URL Manager.

Depending on whether the path requires or accepts parameters, a URL may consist entirely of the path, or it may be extended to include additional information for accessing a specific feature within the main location.

About Parameters

Each URL path may accept one or more parameter, which further defines the feature or item that will be accessed by the URL.

In some cases, the path does not accept any parameters. This is the case when the path is itself associated with a specific location; no additional information is needed in order to open the specified feature. For example, the URL **meridium://Inspection**, which consists entirely of the path with no parameters, is associated specifically with the **Inspection Management Start Page**. A link constructed from this URL will open the **Inspection Management Start Page**. No additional information is needed.

In other cases, parameters are required. For example, the path **Registry/ReportViewer** requires parameters. This is because the path alone does not provide the system with enough information to access the particular feature. In this case, where the Report Viewer is a feature that lets you view existing reports, the address requires parameters that indicate which specific report you want to view. Without this information, the link constructed using only the path, **meridium://Registry/ReportViewer**, will generate an error, indicating that a report was not specified.

In still other cases, parameters are optional. This means that it is possible to provide additional information to access a more specific location but that if specific information is not provided, the system will access the root location without generating errors.

The Syntax of Parameters

The following URL will launch a KPI in the **View KPI**page:

meridium://Scorecard/KPI?Page=View&EntityKey=2330513

This example contains two parameters:

- **Page:** Specifies whether the KPI will be launched in the **View KPI** page or the **Edit KPI** page.
- **EntityKey:** The Entity Key associated with the KPI that will be opened.

Note that each parameter consists of two parts: the parameter name and a value. The parameter name is separated from the value with an equals sign. So, in this example, the name/value pairs are:

- **Page=View**, which indicates that the specified KPI will be opened in the **View KPI** page. Specifying the value **Edit** for the **Page** parameter would open the KPI in the **Edit KPI** page.
- **EntityKey=2330513**, which specifies that the URL will open the KPI with the Entity Key 2330513.

> **Note:** Some parameters allow you to specify multiple values, in which case you would separate each value with a comma. For example, if the **EntityKey** parameter accepted multiple values, you could specify **EntityKey=2330513,2354784,2456781** to access multiple entities.

Finally, note the following about this example:

- Parameters are separated from the path by a **?** (question mark). When constructing URLs, be sure to insert a question mark after the path, before the parameters.
- When a URL contains multiple parameters, the parameter/value pairs are separated by an **&** (ampersand).

Using Keys as Parameter Values

Every stored Meridium APM object has an internal system-assigned numeric *key* that identifies the item within the system. For example, when you create a graph and save it to the Catalog, Meridium APM assigns a key to that graph. Many URL parameters accept keys as values to let you access specific items. For example, if a dataset is assigned the key 1234567, the following URL would open that dataset:

meridium://Registry/Dataset?Key=1234567

Different URLs support the use of different keys, depending on what they are designed to access. The following keys are used most commonly in URLs:

- **Family Key (FMLY_KEY):** The numeric value assigned to an entity or relationship family.
- **Entity Key (ENTY_KEY):** The numeric value assigned to a specific record.
- **Relationship Definition Key (RLDF_KEY):** The numeric value assigned to a relationship definition.
- **Catalog Item Key:** The numeric value assigned to a Catalog item (e.g., saved search, query, report, dataset, or graph).

About Variable Parameter Values

Depending upon the type of URL you construct and where you put the link, you may want to specify a *variable* parameter value instead of an actual value.

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 syntax for specifying a variable parameter value is slightly different, depending upon where you are constructing the URL. You will need to use different syntax for:

- Specifying a variable parameter value in a URL added to a query.
- Specifying a variable parameter value in an Associated Page that is constructed in the URL Manager.

About the URL Builder

The **URL Builder** can assist you in building simple or complex URLs for links to both internal and external locations. You can access the **URL Builder** from the Home Page, a query, and most places where you are allowed to insert hyperlinks.

You will know that you can access the **URL Builder** whenever you see the ⊡ button in a URL field. For example, you can access the **URL Builder** by clicking the ⊡ button in the **URL** field on the **New Hyperlink** dialog box, which is accessible from the Home Page.
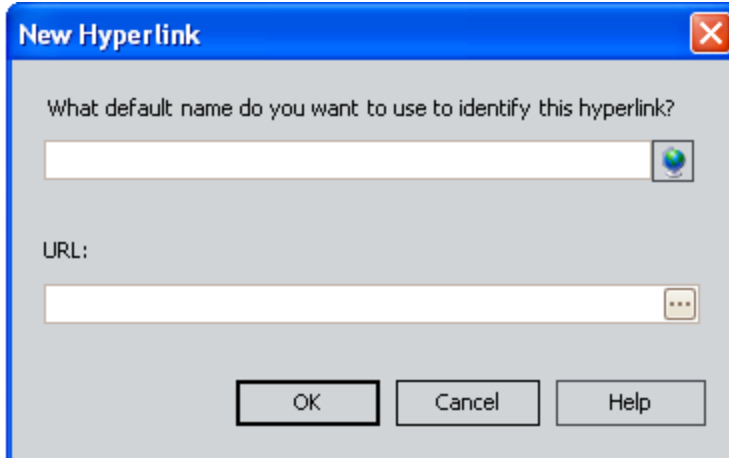
If you want to build a URL to an internal location, the **URL Builder** provides a list of labels for all URLs that have been registered via the URL Manager. In most cases, selecting a label will direct you to a screen where you can specify parameters to the associated URL. In these cases, you should be familiar with URL syntax, as well as the parameters and values accepted by the URL for the specified module. In other cases, depending on the module for which you want to construct a URL, the **URL Builder** will guide you through the process of specifying parameters.

Note that throughout our discussion of the **URL Builder**, the images and descriptions we provide of the baseline labels and URLs will only apply if you have chosen to keep the baseline URLs and corresponding labels. If you modify the URLs or the labels, the explanations we provide may or may not apply.
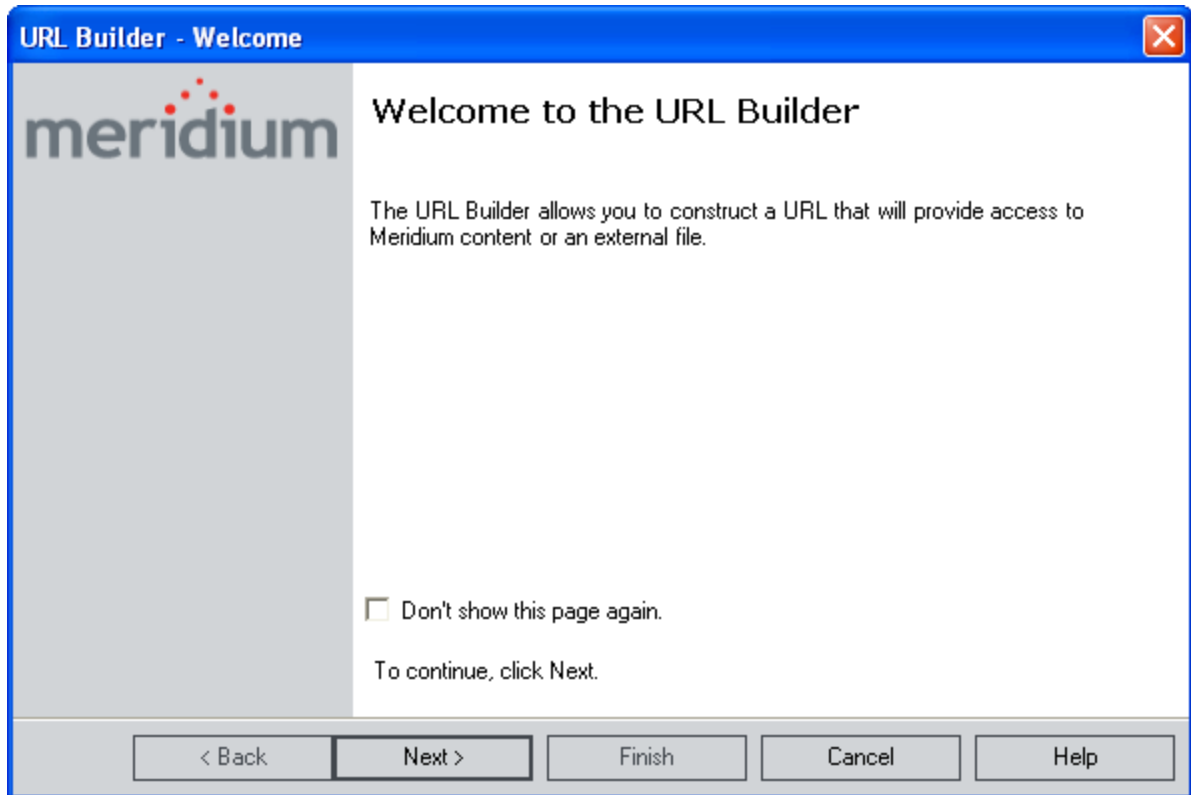
Building a URL for an Internal Location

You can use the **URL Builder** to construct a link to any item in the Meridium APM Framework application or any [external location](#). These instructions provide details on constructing an internal link.
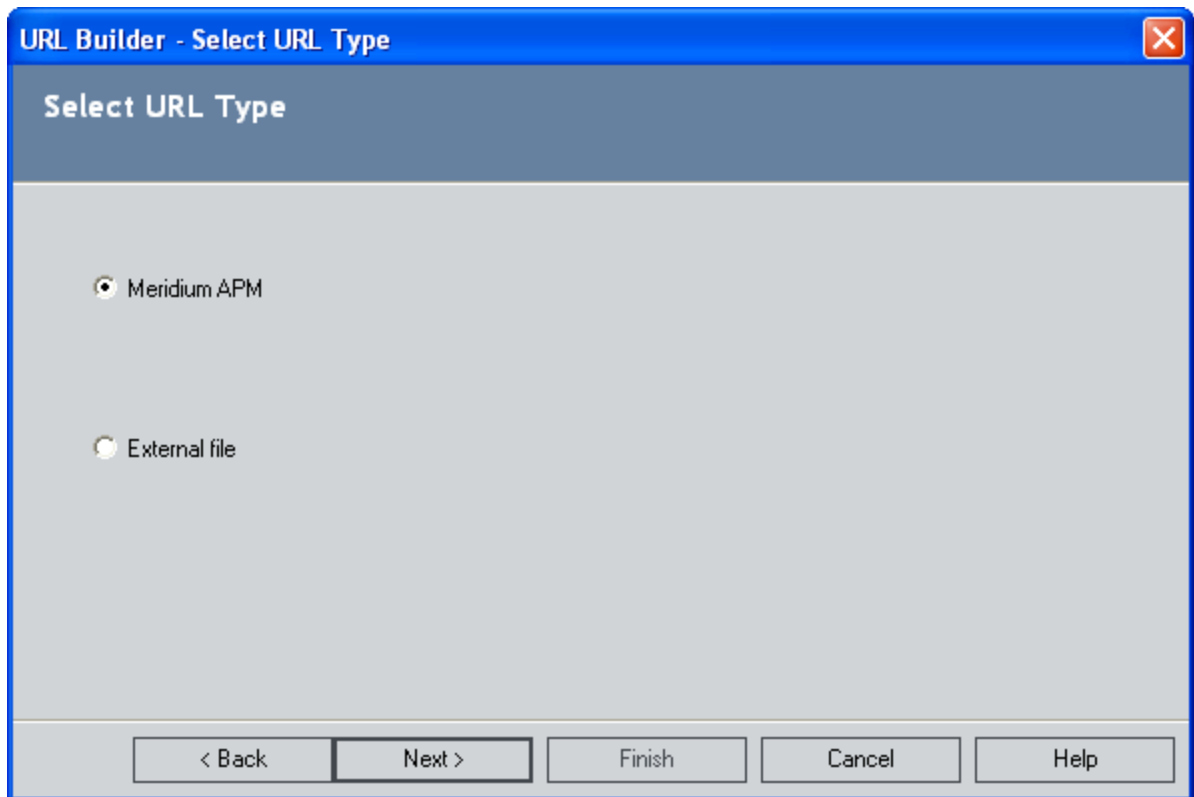
To construct a URL using the URL Builder:

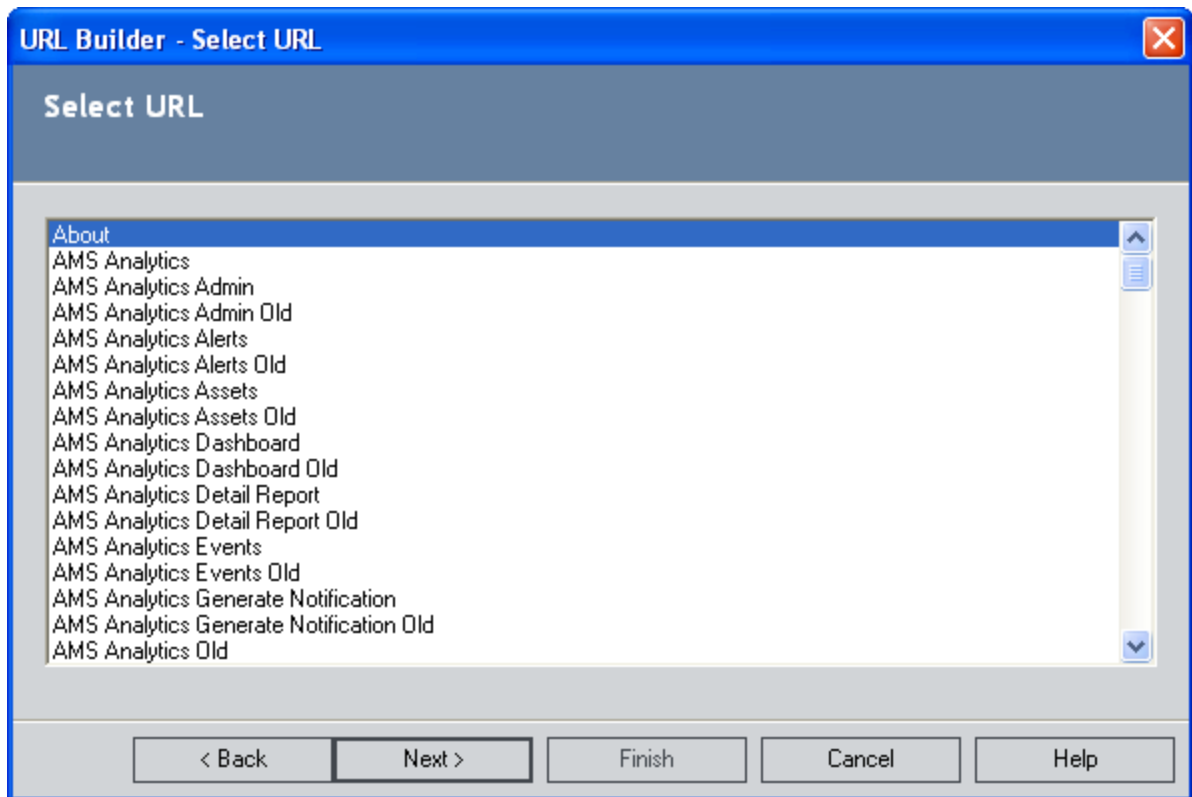1. [Access the **URL Builder**](#).

   The **Welcome** screen appears.

2. Click the **Next** button.

   The **Select URL Type** screen appears.

3. Accept the default selection, **Meridium APM**, and click the **Next** button.

The **Select URL** screen appears.

4. Scroll through the list, and select the type of URL that you want to create. If you want to create a URL for one of the following options, the **URL Builder** will provide detailed steps to guide you through the URL creation.

- A search (link to a new searchor an existing search)
- Record Manager (to create a new recordor open an existing record)
- Catalog item
- Any Reliability analysis (i.e., Reliability Automation, Reliability Cost, Reliability Distribution, Reliability Distribution General, Reliability Growth, Reliability Systems Analysis, or Reliability Systems Model)
- Event Builder
- Task Builder

If you choose a URL type for which detailed steps are not available, the Base URL is created, and you can specify parameters as needed, if appropriate.

5. Click the **Next** button.

6. Complete the remaining steps of the process.

7. Click the **Finish** button.

The URL is created.

Building a URL for an External Location

You can use the **URL Builder** to construct a link to any item in the Meridium APM Framework application or any external location. These instructions provide details on constructing an external link.
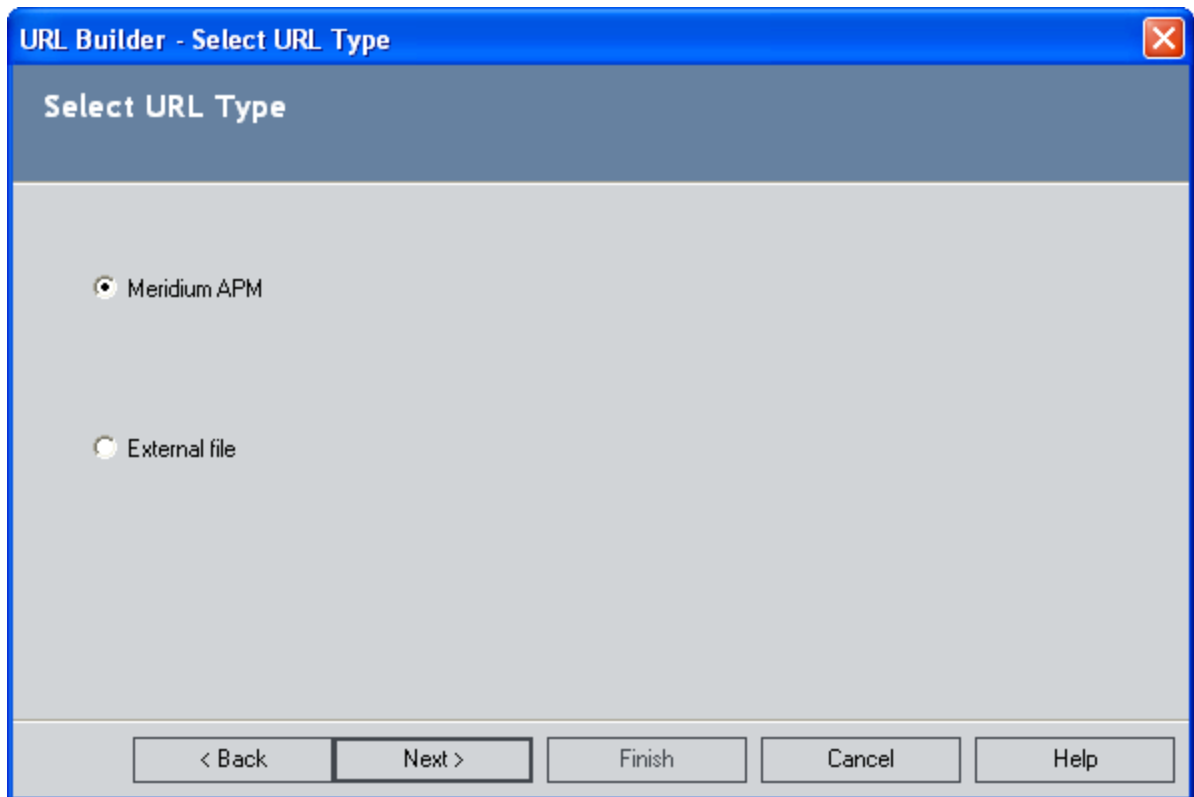
To construct a URL using the URL Builder:

1. Access the URL Builder.

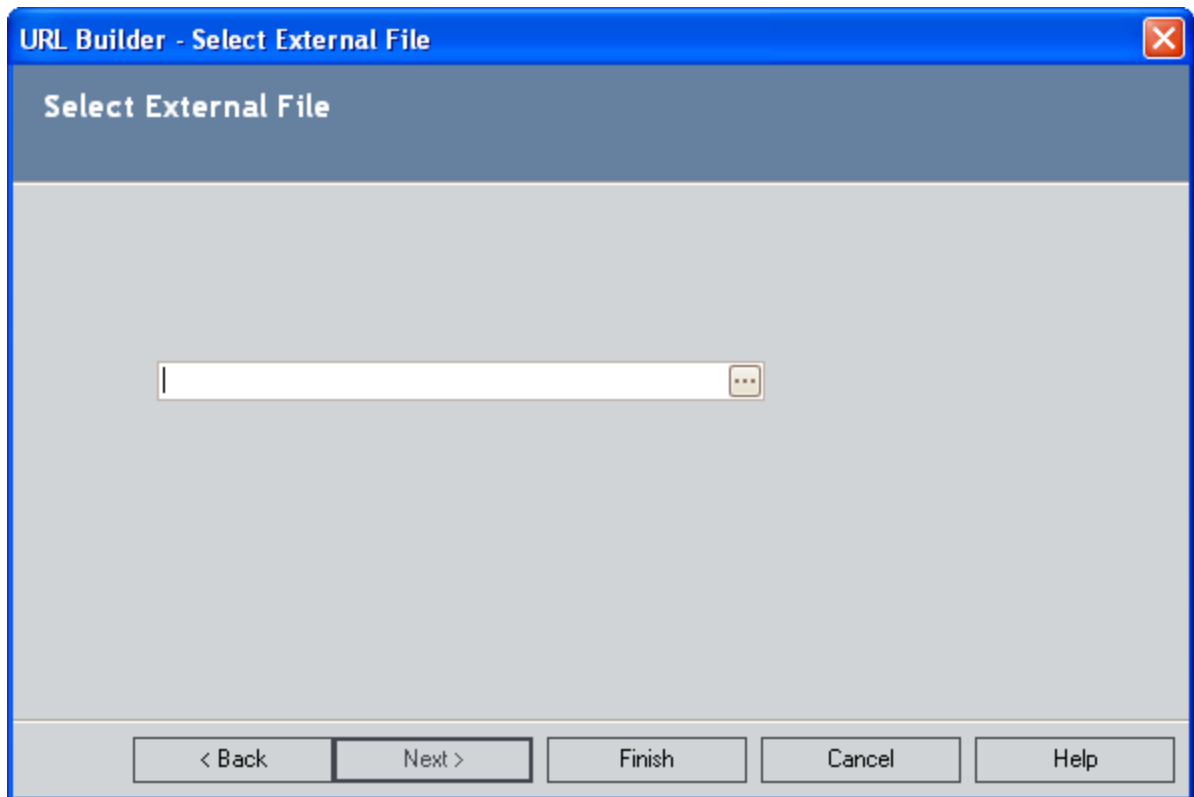   The **Welcome** screen appears.
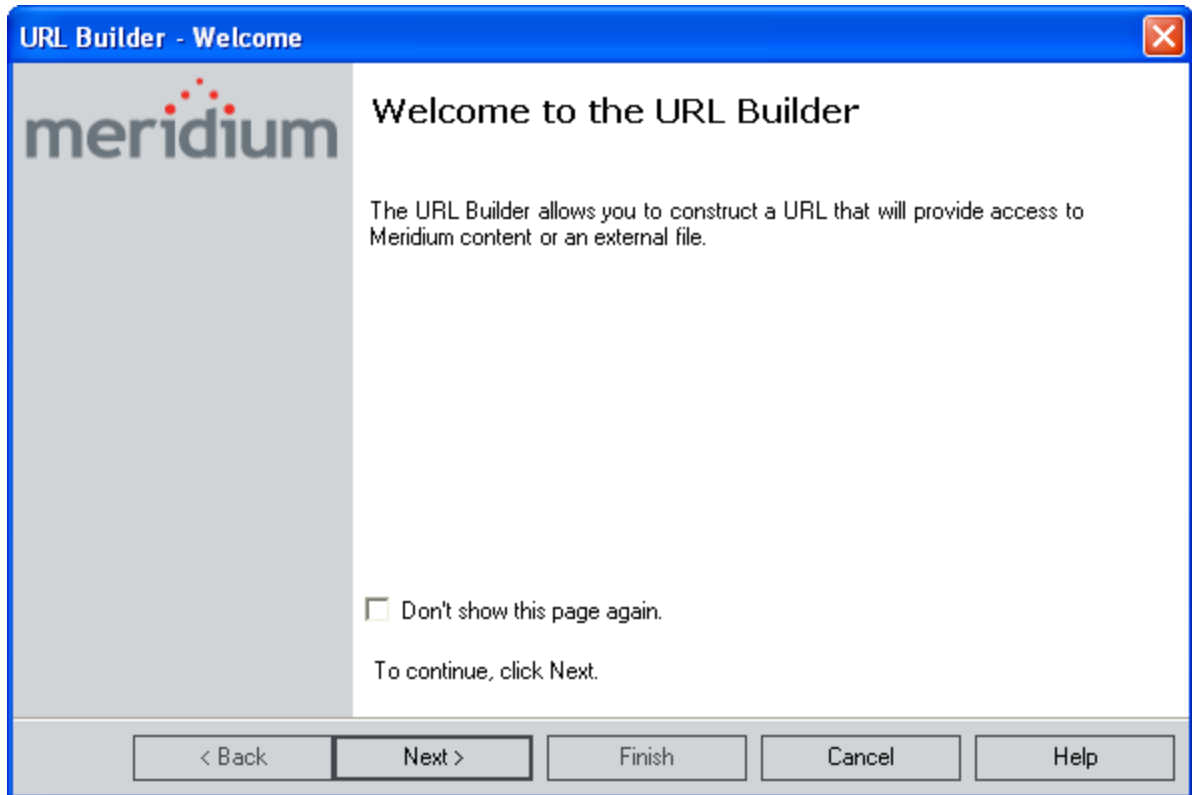
2. Click the **Next** button.

   The **Select URL Type** screen appears.

3. Select the **External File** option, and click the **Next** button.

The **Select External File** screen appears.

4. Click the ⋯ button.

   The **Select File** dialog box appears.

5. Select the file to which you want to link, and click the **Open** button.

6. In the **URL Builder**, click the **Finish** button.

   The URL is created.