



Meridium APM Policy Manager

3.6.1.1.0



Meridium APM Policy Manager

3.6.1.1.0

Copyright © Meridium, Inc. 2017

All rights reserved. Printed in the U.S.A.

This software/documentation contains proprietary information of Meridium, Inc.; it is provided under a license agreement containing restrictions on use and disclosure. All rights including reproduction by photographic or electronic process and translation into other languages of this material are fully reserved under copyright laws. Reproduction or use of this material in whole or in part in any manner without written permission from Meridium, Inc. is strictly prohibited.

Meridium is a registered trademark of Meridium, Inc.

All trade names referenced are the service mark, trademark or registered trademark of the respective manufacturer.

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.meridium.com/documentation/WebHelp/WebHelpMaster.htm>).

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

Table of Contents

Meridium APM Policy Manager	1
Copyright and Legal	2
About This Document	3
Table of Contents	4
Overview of Policy Manager	10
Policy Manager System Requirements	11
Policy Manager Data Model	17
About the Date Format Used in Policy Manager	19
About Policies	20
About Policy Instances	21
Putting It All Together: Policies and Instances	22
About These Examples	24
Example Policy: Monitoring a Single Value	25
Example Policy: Monitoring Multiple Values	26
Example Policy: Creating Health Indicator and Health Indicator Value Records	27
Example Policy: Creating and Updating a Policy Event Record	29
First-Time Deployment Workflow	30
Upgrade or Update Policy Manager to 3.6.1.1.0	32
How are Policies Executed?	45
About Configuring Policies to be Executed When Records or Tag Values are Updated	46
About Configuring Policies to be Executed on a Predefined Schedule	49
Installing the Meridium APM Policies Execution Service	50
Modifying the File Meridium.Policies.ExecutionService.exe.config	62
Modifying the File Meridium.Policies.Service.exe.config	64
Creating the Scheduled Item to Execute Policies on a Predefined Schedule	66
Policy Manager Security Groups	70
Policy Manager Workflow	71
Using the Site Map in Policy Manager	72

Accessing the Policy Manager Start Page	73
Aspects of the Policy Manager Start Page	74
Policy Manager Section	75
My Recent Policies Section	76
Policy Executions Section	77
Policy Results Section	78
Policy Recommendations Section	79
My Policy Recommendations Section	80
Accessing the Policy Page	81
Aspects of the Policy Page	82
Policy Actions Toolbar	84
Inputs Toolbar	85
Conditions Toolbar	86
Logic Toolbar	87
Calculations Toolbar	88
Actions Toolbar	89
About the Policy Diagram	90
About Tabs Displayed on the Policy Page	91
The Search Area	92
The Details Area	93
Policy Details Tab	94
Execution Settings Tab	95
Instances Tab	97
Execution History Tab	101
Validation Tab	102
Types of Messages in Policy Manager	104
Validation Messages	105
System Messages	106
Error Messages	107
Refreshing the Policy Page	108

About the Zoom Tool	110
Creating Policies from Scratch	112
Copying a Policy	114
Copying a Policy and its Associated Instances	115
Opening Existing Policies	116
Aspects of the Policies Search Page	118
About Policy Ownership	120
Taking Ownership of a Policy	121
Viewing Policy Ownership History	123
Saving Changes to a Policy	124
Activating Policies	125
Deactivating Policies	126
Deleting Policies	127
About the Time Zone Setting	128
Policy Diagram Workflow	131
About Working with the Policy Diagram	132
Adding Nodes to the Policy Diagram	134
Connecting Nodes in a Diagram	135
Configuring Node Properties	136
About Node Names and Labels	137
Defining Input Values	140
Input Values Example	142
About Connection Properties	143
Copying and Pasting Nodes and Connections	145
About Input Nodes	147
About the Trigger Check Box	148
Entity Nodes	149
Query Nodes	151
Point Value Nodes	155
Measurement Location Nodes	157

AMS Asset Nodes	160
User Nodes	162
Constant Nodes	165
Health Indicator Nodes	167
Xi Tag Nodes	170
GE Tag Nodes	173
Comparison Nodes	175
Case Nodes	178
Using the Case Node to Change Output Values	182
Using the Case Node to Combine Policy Logic	184
Logic Nodes	186
About Calculation Nodes	189
Add, Subtract, Multiply, Divide, Remainder, and Exponent Nodes	190
Round Nodes	194
R-Script Nodes	198
Collection Filter Nodes	201
Last Nodes	204
Min and Max Nodes	206
Average Nodes	209
Sum Nodes	211
Count Nodes	213
Threshold Statistics Nodes	215
About Action Nodes	220
Email Contact Nodes	221
Create Event Nodes	225
Close Event Nodes	231
Create Recommendation Nodes	234
Rule Nodes	238
Return Value Nodes	239
Deactivate This Instance Nodes	240

Add Value to Health Indicator Nodes	242
Create Production Event Nodes	244
About Specifying Dates and Times to Evaluate	246
Constants	248
What is the Difference Between Start and Now?	250
Variables	252
About Specifying an Amount of Time to Evaluate	253
Example Diagram: Monitoring a Single Value	256
Example Diagram: Monitoring Multiple Values	257
Example Diagram: Creating Health Indicator and Health Indicator Value Records	260
Example Diagram: Creating and Updating a Policy Event Record	263
Primary Records and Primary Nodes	265
Workflow for Identifying Primary Records	267
Example: Primary Node and Primary Record	268
Example: Primary Record Only	270
Identifying the Primary Node in a Policy	272
Specifying the Primary Record in a Policy when a Primary Node is Identified	274
Specifying the Primary Record in a Policy when a Primary Node is Not Identified	276
Creating Policy Instances	278
Assigning Specific Records to Nodes	280
Activating a Single Policy Instance	283
Activating All Policy Instances Associated with a Policy	284
Deactivating a Single Policy Instance	285
Deactivating All Policy Instances Associated with a Policy	286
Creating a New Health Indicator Record from the Instances Tab	287
Saving Changes to a Policy Instance	292
Deleting Policy Instances	293
About Validating the Policy Logic	294
Validating the Logic Using Manual Values	295
Validating the Logic Using Policy Instance Values	296

Copying Policy Instance Values for Validation	297
About Configuring Policies to be Executed	299
Configuring Policies to Be Executed When Records Are Updated	300
Configuring Policies to Be Executed on a Predefined Schedule	301
About the Execution Results	305
Interpreting Node Color-Coding in the Policy Diagram	307
Aspects of the Execution Details Windows	308
Dashboards Folder	310
Queries Folder	311
Dashboard Queries Folder	312
Policy Manager Families Delivered with Read-Only Properties	313
Policy	314
Policy Event	318
Policy Instance	321
Policy Recommendation	323
Parameters for the Policy Manager URL	327
Examples of Policy Manager URLs	328
Policy	329
Policy Diagram	330
Policy Instance	331
Primary Record	332
Primary Node	333

Overview of Policy Manager

Traditionally, many companies attempt to improve equipment and location performance by identifying historical trends and using them to define future actions. For example, the results of a Reliability Distribution Analysis may indicate that a type of equipment fails about every six months, so a strategy might be implemented to replace equipment of that type every five months (i.e., one month prior to the next predicted failure). In reality, however, *some* equipment of that type *could* last longer than six months (e.g., for eight months), and in these cases, replacing the equipment too early would result in unnecessary costs. Waiting until the equipment fails, however, is also not the best plan.

The objective of the *Policy Manager* module is to develop and execute strategies based upon historical data and dynamic data, which is obtained by monitoring conditions in real time. Using Policy Manager, you can [define policies](#) that:

- Retrieve information about current conditions.
- Allow you to identify emerging trends of equipment and locations.
- Apply some logic to that data based on historical trends.
- Initiate actions that will mitigate future risk.

For example, an engineer might know that when a motor reaches 200 degrees, it will begin to fail. Historical data might also tell him that the same motor operating continuously will overheat about every three months. Rather than implementing a strategy to replace the motor every three months, as suggested by the historical data, he can create a policy to take action when threshold temperature conditions are *actually* met. If the policy retrieves a reading from a process historian indicating that the motor has reached 200 degrees, policy logic can create a recommendation to replace the equipment automatically. In this way, the company can save money by avoiding unnecessary replacements and *still* prevent failures that will occur if the equipment is replaced too late.

Additionally, after a policy has been created, it can monitor current conditions continuously and automatically assign appropriate actions to any employees, based upon the criteria that have been defined by the policy owner. In this way, even employees who do not have the expertise to analyze real-time data can implement strategies based upon emerging trends. In other words, policies give you the ability to conduct automatic analysis of real-time data and implemented appropriate actions quickly and reliably.

Policy Manager System Requirements

The *Policy Manager* license is required to take advantage of Policy Manager functionality. In addition, your system must contain the basic Meridium APM system architecture and the following additional components:

- **Meridium APM Policy Execution Server:** The machine that houses the Meridium Policies Execution Service. The Meridium Policies Execution Service works in combination with the Meridium Policies Trigger Service to execute policies. The Meridium Policies Trigger Service is installed automatically on the Meridium APM Application Server machine and the Meridium APM Process Data Integration Server.

The Meridium Policies Execution Service is installed by the Meridium APM Server and Add-ons installer when you select the *Meridium APM Policy Execution Service* option in the installer. A supporting instance of the Meridium APM Application Server and Add-ons should also be installed on the Meridium APM Policy Execution Server.

Depending upon the number of policies that you will need to manage in your system, you may have multiple Meridium APM Policy Execution Server machines. In that case, you will need to perform all the installation and configuration steps on each machine.

About the Policy Execution Server and the Application Server

Policies will be executed according to a defined schedule or on an as-needed basis when the values in the records that are included in the policy change. If the records included in a policy belong to families *other than* the Xi Tag family (e.g., the Equipment family), when the policy is executed, the Meridium Policies Trigger Service on the *Meridium APM Application Server* is used in combination with the Meridium Policies Execution Service on the Meridium APM Policy Execution server.

[Click here to expand and collapse the hardware and software requirements for the Meridium APM Policy Execution Server.](#)

The following hardware is the *minimum* requirement for any machine that will serve as the Meridium APM Policy Execution Server. These requirements account for the Meridium APM software only. Additional hardware may be required on machines where other components are installed. These requirements may not be sufficient, depending upon how your system is configured.

- Two Process Cores, 2+ GHz
- 4 GB RAM
- 20 GB Hard Drive space

The Meridium APM Policy Execution Service requires that the following software components exist on the machine where it will be installed. A supporting instance of the Meridium APM Application Server is also required. [Click here for the APM](#)

Application Server software requirements. This table is intended to be used as a reference of the behavior of the Meridium APM Application Server and Add-ons installer as it relates to each component.

Software Name and Version	Description	Where You Can Obtain It	Behavior of Meridium APM Server and Add-ons Installer
Microsoft.net 3.5 SP1	Microsoft software framework	Microsoft	If it is not found, the installer will close.
Microsoft.net 4.0	Microsoft software framework	Microsoft	If it is not found, the installer will close.
MSMQ	Message queuing service	Microsoft	If it is not found, you can continue through the installer. It can be activated through roles in Server Manager.
Windows 7 or 2008 R2	Microsoft operating system	Microsoft	If it is not found, a message will appear, indicating that the environment is not supported, but you can continue through the installer.
64-bit operating system	Microsoft operating system	Microsoft	If it is not found, the installer will close.
ODP.net 2.112.1.0 or later	Oracle Data Provider	Oracle	If it is not found, a message will appear indicating that if you are using Oracle as your database provider, the software is required, but you can continue through the installer.

Software Name and Version	Description	Where You Can Obtain It	Behavior of Meridium APM Server and Add-ons Installer
Merge Modules: Visual C++ 8 CRT (x86) Win SXS MSM			
Visual C++ 8.0 MFC (x86) WinSXS MSM			
Visual C++ 10.0 CRT (x64)	Third-party merge modules that support C++ runtime.	Meridium APM	If this is not found, the installer will install it automatically.
Visual C++ 9.0 CRT (x86) WinSXS MSM			
Visual C++ 10.0 ATL (x64)			
Visual C++ 9.0 ATL (x86) WinSXS MSM			

- **Meridium APM Process Data Integration Server:** The machine on which the Meridium Policies Trigger Service is installed. This server also contains the OPC Server software and a supporting instance of the Meridium APM Application Server and Add-ons.

This machine is required *only if you want to use Xi Tag records in your policies*. If a policy includes an Xi Tag record, the policy will be executed automatically when the reading value for that tag changes in the process historian. In this case, the Meridium Policies Trigger Service on the Meridium APM Process Data Integration Server is used in combination with the Meridium Policies Execution Service on the

Meridium APM Execution Server. The Meridium Policies Trigger Service is installed by the Meridium APM Server and Add-ons installer when you install the *Process Data Integration Services* software.

After you have installed and configured the basic Meridium APM system architecture, you will need to perform some configuration steps specifically for Policy Manager.

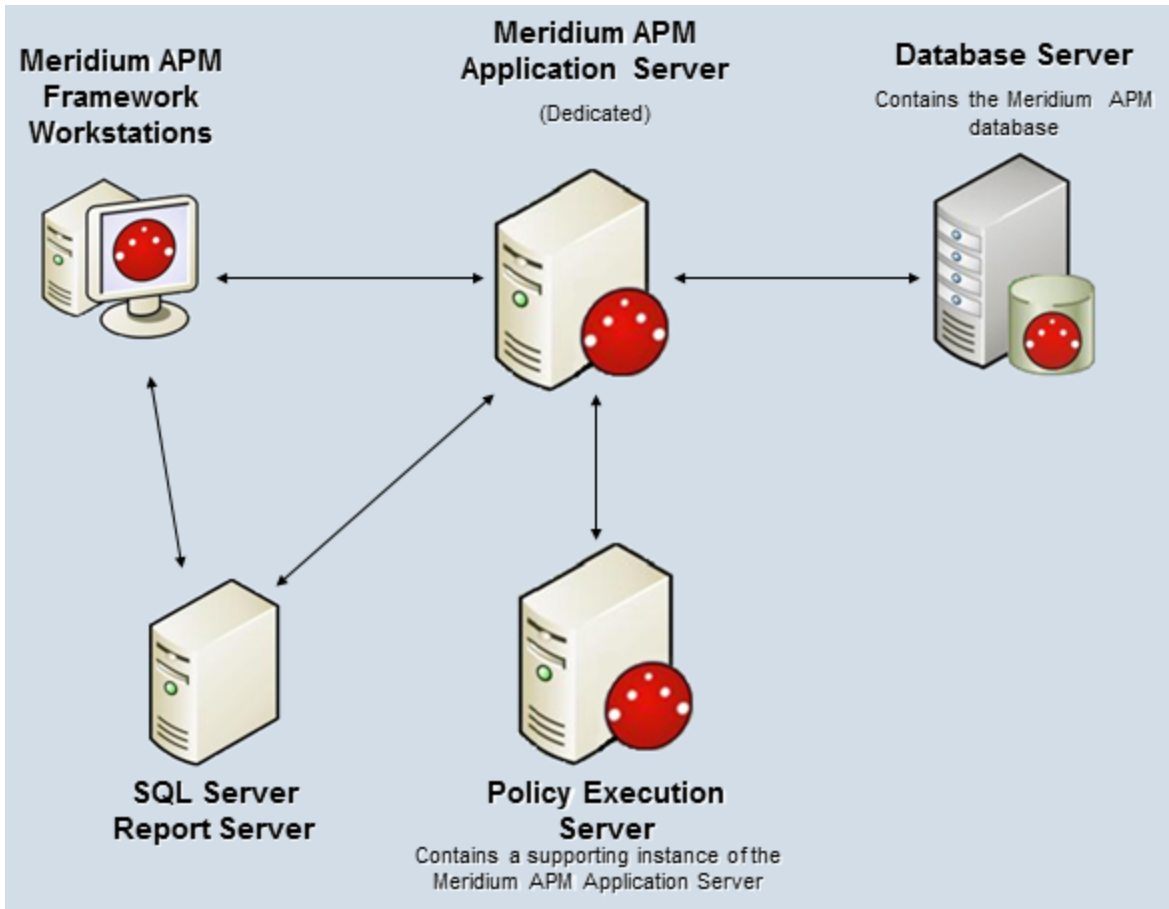
If you are using a virtual machine (VM) to host your Meridium APM Policy Execution Server or your Meridium APM Process Data Integration Server, we recommend that:

- You increase the physical server CPU and hard drive space requirements by 20% to ensure that the application will perform equivalently in a virtualized environment.
- You double the physical server memory requirements.
- Each VM have its own dedicated processor cores and memory allocation to ensure appropriate performance.

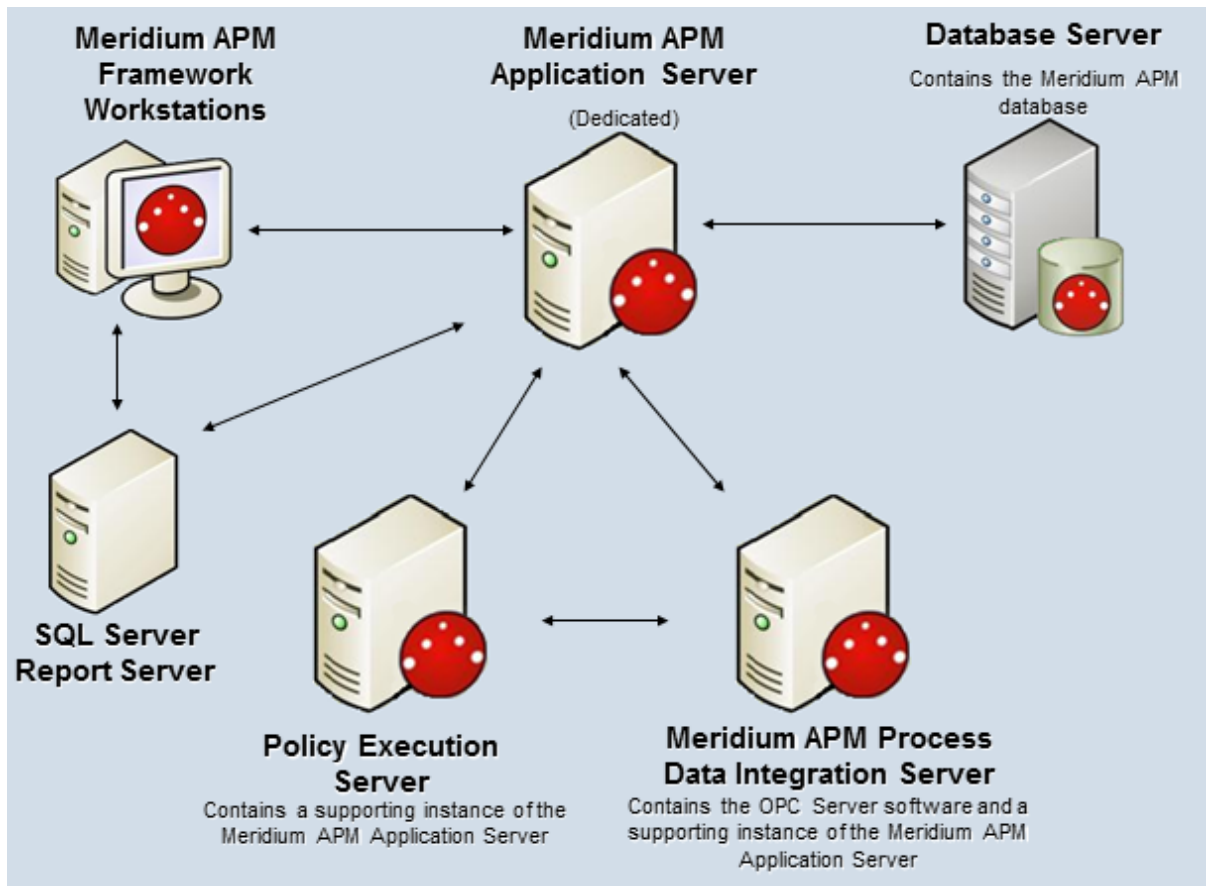
Meridium, Inc. supports Hyper-V and VMWare platforms.

These requirements may not be sufficient, depending upon how your system is configured. Parameters that affect the hardware requirements include the number of users, modules purchased, database size, and other factors that can vary from one customer to another. For help refining your specific system requirements, consult a member of the Meridium Inc. Global Support Services or Professional Services department.

The following image provides an illustration of the architecture components that are required for deploying Policy Manager. This image shows the system architecture for a Policy Manager deployment that does not include Process Data Integration.



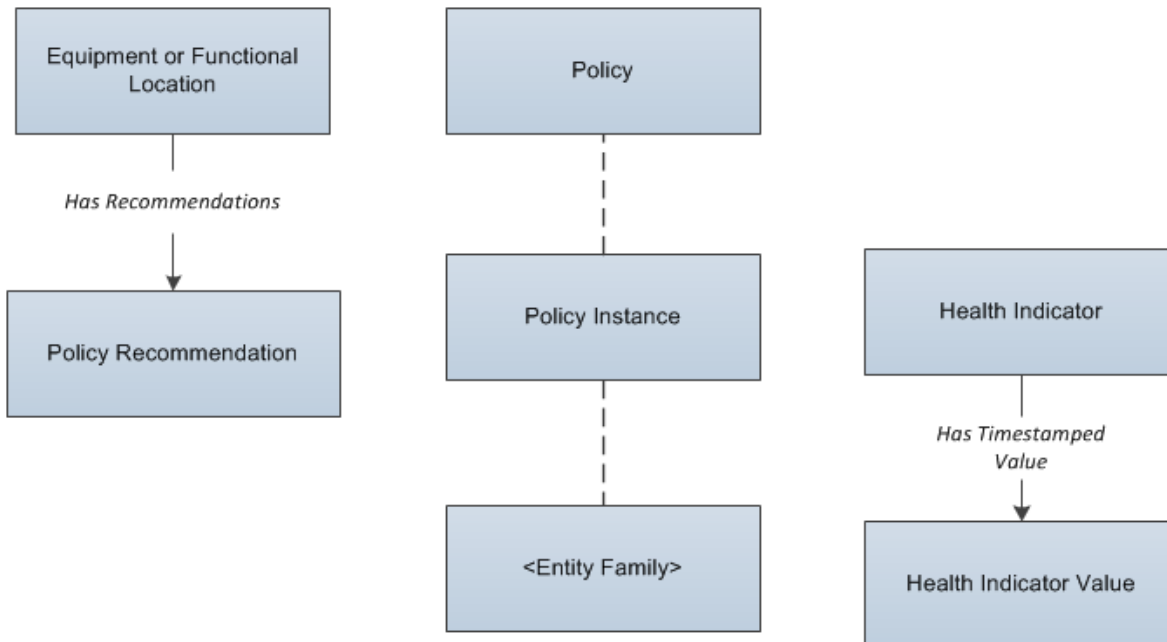
The following image provides an illustration of the architecture components that are required for deploying Policy Manager, including Process Data Integration.



Policy Manager Data Model

The Policy Manager data model uses families to represent the components of policies. In the following diagram:

- Boxes represent entity families.
- Solid lines represent relationship families.
- Dotted lines represent *associations* between records (not relationships).



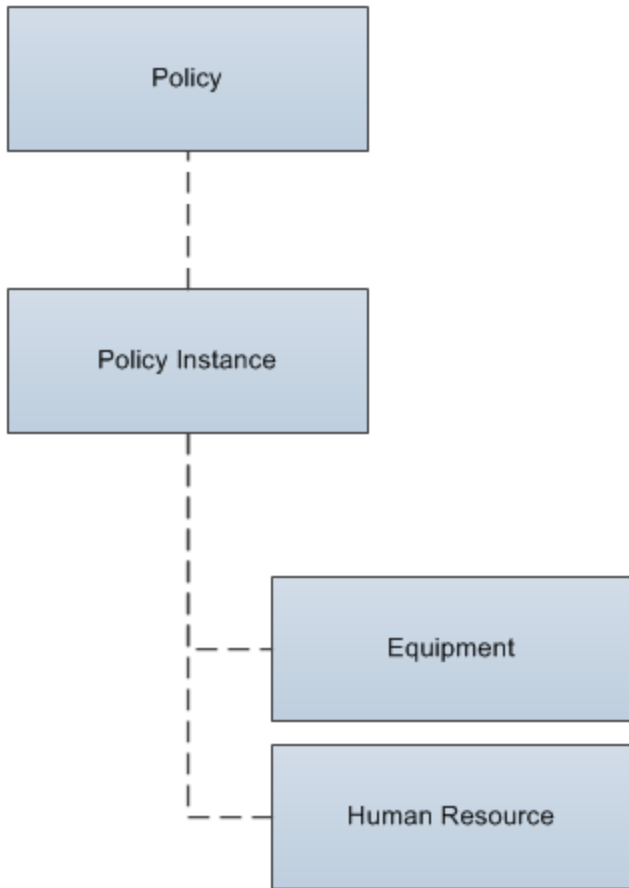
In Policy Manager:

- *Policy records* store identifying information about [policies](#), such as the name, description, and next and last execution dates. The Policy family contains a binary field, *Model*, that stores code defining the logic that is represented by the policy diagram.
- *Policy Instance* records store information about [policy instances](#). The Policy Instance family contains multiple binary fields that store code defining the records that belong to the policy instance.

In the data model image

- The Health Indicator, Health Indicator Value, and Has Timestamped Value families are used only when you [include an Add Value to Health Indicator node in a policy](#).
- The *<Entity Family>* box represents any family containing a record that is included in the policy instance. For example, if a policy diagram contains an *Equipment* node and a *Human Resource* node, a policy instance that is associated with that policy would include two records: one that is assigned to the *Equipment* node and

one that is assigned to the *Human Resource* node. So, the data structure would look like this:



About the Date Format Used in Policy Manager

With the exception of fields that accept a [relative date format](#), the date format *YYYY-MM-DD hh: mm: ss* is used for *all* date fields in all families included in the [Policy Manager data model](#). This format is used regardless of any global date format settings that have been defined or the format that is defined in your Windows Region and Language settings.

About Policies

Policies are strategic plans that define actions that should be taken when certain conditions exist. To build a policy, you will use a diagram, where the nodes on the diagram represent:

- The items that you want to monitor (e.g., equipment temperature).
- The conditions that should trigger actions to be taken (e.g., temperature rises above 600 degrees).
- The actions that should be taken (e.g., send an email message).

Policies do not reference specific *records*. Rather, they can contain references to *families*. After you have designed your policy, you can add *policy instances*, which identify the individual records whose values will be evaluated against the criteria that is specified in the policy. Each policy can have an unlimited number of policy instances.

The concept of policies and policy instances is best understood through a [complete, simple example](#).

About Policy Instances

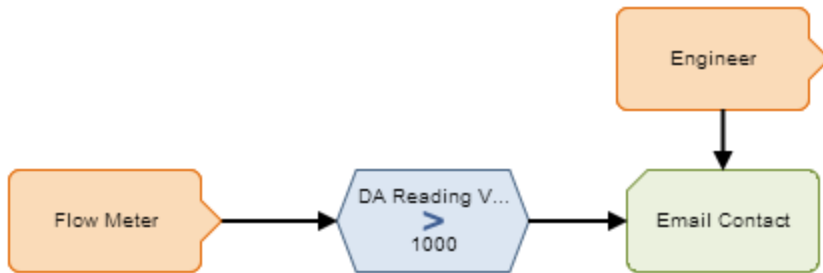
Each [policy](#) can be associated with one or more *policy instances*, where each policy instance identifies the individual records you want to monitor (e.g., Equipment 101). The conditions that should trigger actions to be taken (e.g., temperature rises above 600 degrees) and the specific actions that should be taken (e.g., send an email message to Bob Smith) are defined by the [policy](#) itself using the policy diagram.

In other words, using a policy instance, you will identify the specific records whose values should be used as inputs to the policy logic when the policy is executed.

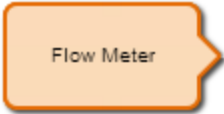
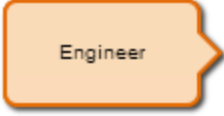
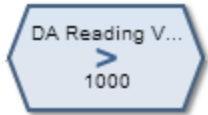
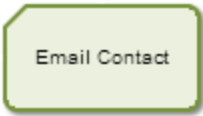
After you have created a policy instance, you can [validate the policy logic](#) against the records included in that policy instance to ensure the logic is working correctly. After the policy is executed, you can [view execution results](#) for each policy instance.

Putting It All Together: Policies and Instances

Consider a policy that monitors the DA Reading Value related to an Xi Tag record. The following image displays the policy diagram.



This policy diagram contains the following nodes.

Node Type	Node	Purpose
Input		Serves as a placeholder for specific Xi Tag records that will be included in individual policy instances belonging to this policy.
Input		Serves as a placeholder for specific Human Resource records that will be included in individual policy instances belonging to this policy.
Condition		Defines the logic that will be applied to values associated with specific Xi Tag records that will be included in individual policy instances belonging to this policy.
Action		Specifies that an email message should be sent when the condition defined in the condition node is met. The email message will be sent to the email address defined in the specific Human Resource records that will be included in individual policy instances belonging to this policy.

A policy instance associated with this policy might look like this:

Source	Instance Value
★ Flow Meter	A1113B
Engineer	Smith, John

...where:

- The specific Xi Tag record is *A1113B*.
- The specific Human Resource record contains the last and first name *Smith, John*.

When this policy is executed, if the Xi Tag record *A1113B* has a DA Reading Value that is greater than *1000*, the Meridium APM system will send an email message to the email address specified in the Human Resource record with the last and first name *Smith, John*.

[Click here](#) for additional information about working with the policy diagram, including further explanation of this example diagram.

About These Examples

To explain how you can use Policy Manager to accomplish certain tasks, we have created the following policy examples:

- A policy that [monitors a single value](#) (temperature).
- A policy that [monitors multiple values](#) (vibration, pressure, temperature, and flow).
- A policy that [creates a Health Indicator record and a Health Indicator Value record](#).
- A policy that [creates and updates a Policy Event record](#).

Example Policy: Monitoring a Single Value

Suppose a factory contains a heat exchanger shell, which can fail as a result of hot hydrogen attack. Part of the mitigation strategy for the heat exchanger shell, therefore, is to monitor the temperature of the process stream exposed to the shell component to ensure that it does not exceed 600 degrees Fahrenheit.

The following table outlines a workflow that would yield a policy that monitors the process steam temperature and triggers certain actions if it gets too high. Because the policy uses inputs from other modules, the **Location** column identifies the Meridium APM module or feature in which you would complete a step that yields values that will be used by those policy inputs.

Step	Location	Task
1	RBI	Assess the risk of loss of containment for a heat exchanger shell, which is associated with the damage mechanism <i>Hot Hydrogen Attack</i> .
2	Process Data Integration	Create an Xi Tag record for the location on the heat exchanger shell where the process stream temperature is monitored and recorded (using an external process historian).
3	AHM	Create a Health Indicator record based on the Xi Tag record.
4	Policy Manager	Create a policy to monitor the temperature values that are stored in the process historian.

The policy that you create could specify that, when the policy is executed, if the temperature of the heat exchanger shell rises above 600 degrees Fahrenheit more than once:

- A Policy Recommendation record will be created and configured to create an Oracle EBS eAM Work Request for immediate corrective action.
- An email message will be sent to a process engineer.

[Click here](#) for detailed information about a policy diagram that illustrates this example.

Example Policy: Monitoring Multiple Values

Suppose a factory contains a highly critical pump, which can pose risk if one of its bearings fails. Part of the mitigation strategy for the pump, therefore, is to perform a vibration analysis and monitor multiple variables in order to create a composite view of the pump's condition.

The following table outlines a workflow that would yield a policy that monitors the pump's condition and triggers certain actions if impending failure is identified. Because the policy uses inputs from other modules, the **Location** column identifies the Meridium APM module or feature in which you would complete a step that yields values that will be used by those policy inputs.

Step	Location	Task
1	ASM	Assess the risk of bearing failure for the pump.
2	Process Data Integration	Create Xi Tag records for the locations on the pump where the vibration, pressure, temperature, and flow are monitored and recorded (using an external process historian).
3	AHM	Create Health Indicator records based on the Xi Tag records.
4	Policy Manager	Create a policy to monitor the values that are stored in the process historian.

The policy that you create could specify that, when the policy is executed, if the pump conditions indicate a pending failure:

- A Policy Recommendation record will be created and configured to create an Oracle EBS eAM Work Request for immediate corrective action.
- The reliability engineer will notified via email.

[Click here](#) for detailed information about a policy diagram that illustrates this example.

Example Policy: Creating Health Indicator and Health Indicator Value Records

Unlike the single input and multiple input examples, the policy in this example does not result in corrective actions when thresholds are met. Instead, this example is manipulating multiple readings to create a health index score that will indicate the overall health of the asset.

Suppose you are using Asset Health Manager to monitor the health of a pump using the following indicators, which are tracked using Health Indicator records whose primary source records are Measurement Location records:

- Flow Rate
- Pressure
- Lubrication
- General operation
- Vibration

You can use Asset Health Manager to evaluate the health of the pump in terms of these individual indicators by viewing gauges and trends for them on an individual basis. But you might also want to evaluate the overall health of the pump as a *composite* view of all of these underlying factors. If so, you can create a policy that facilitates this.

The following table outlines a workflow that would yield this type of policy. Because the policy uses inputs from other modules, the **Location** column identifies the Meridium APM module or feature in which you would complete a step that yields values that will be used by those policy inputs.

Step	Location	Task
1	Operator Rounds	Create the Measurement Location records that will be used to measure the pump's conditions. Complete the Operator Rounds workflow to yield reading values for those checkpoints.
2	AHM	Create Health Indicator records based on the Measurement Location records.
3	Policy Manager	Create a policy to monitor the values that are stored in the Health Indicator records.

The policy that you create could specify that, when the policy is executed, the values in the existing Health Indicator records will be used in calculations that determine an overall, *single* value that provides a composite view of the pump's health. This value is stored in a Health Indicator Value record that gets created automatically each time the policy is executed and linked to a Health Indicator record that will be used to evaluate the overall health of the pump.

Example Policy: Creating Health Indicator and Health Indicator Value Records

[Click here](#) for detailed information about a policy diagram that illustrates this example.

Example Policy: Creating and Updating a Policy Event Record

Suppose a factory contains a heat exchanger, which can fail as a result of hot hydrogen attack. Part of the mitigation strategy for the heat exchanger shell, therefore, is to monitor the temperature of the process stream exposed to the shell component to ensure that it does not exceed 600 degrees Fahrenheit. Specifically, the organizational policy dictates that when the temperature of the process stream reaches 550 degrees Fahrenheit, the heat exchanger should be shut down to ensure that the temperature of the process stream exposed to the shell component does not exceed 600 degrees Fahrenheit, and when the temperature of the process stream falls back below 550 degrees Fahrenheit, the heat exchanger should be restarted.

The following table outlines a workflow that would yield a policy that monitors the process steam temperature and triggers the shutdown of the heat exchanger if the process stream temperature reaches 550 degrees Fahrenheit. Because the policy uses inputs from other modules, the **Location** column identifies the Meridium APM module or feature in which you would complete a step that yields values that will be used by those policy inputs.

Step	Location	Task
1	RBI	Assess the risk of loss of containment for a heat exchanger shell, which is associated with the damage mechanism <i>Hot Hydrogen Attack</i> .
2	Process Data Integration	Create an Xi Tag record for the location on the heat exchanger shell where the process stream temperature is monitored and recorded (using an external process historian).
3	AHM	Create a Health Indicator record based on the Xi Tag record.
4	Policy Manager	Create a policy that monitors the temperature values that are stored in the process historian and creates and updates a Policy Event record.

The policy that you create could specify that, when the policy is executed, if the temperature of the process stream exposed to the exchanger shell:

- Rises to or above 550 degrees Fahrenheit, a Policy Event record will be created.
- Falls back below 550 degrees Fahrenheit (after rising to or above 550 degrees Fahrenheit), the Policy Event record will be updated to indicate that the event has closed.

[Click here](#) for detailed information about a policy diagram that illustrates this example.

First-Time Deployment Workflow

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

Whether a step is required or optional is indicated in the **Required/Optional** cell. Steps are marked as *Required* if you must perform the step to take advantage of Policy Manager functionality.

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

Step	Task	Required/Optional
1	<p>Install the Meridium APM Policies Execution Service on the Meridium APM Policy Execution Server.</p> <p>This documentation assumes that the Meridium APM Policy Execution Server machine meets the hardware and software requirements.</p>	Required
2	<p>Modify the Policies Execution Service configuration file to specify the Meridium APM Application Server, Meridium APM Execution Server, Meridium APM database, and login credentials.</p>	Required
3	<p>If it is not already started, start the Meridium APM Policies Execution Service.</p>	Required
4	<p>Modify the Policies Trigger Service configuration file to specify the Meridium APM Application Server, Meridium APM Execution Server, Meridium APM database, and login credentials.</p> <p>You will need to perform this step on the Meridium APM Application Server, the Meridium APM Policy Execution Server, <i>and</i> the Meridium APM Process Data Integration Server.</p>	<p>Always required on the Application Server and Policy Execution Server.</p> <p>Required on the Process Data Integration Server only if you want to use Xi Tag records in your policies.</p>

Step	Task	Required/Optional
5	<p>If it is not already started, start the Meridium APM Policies Trigger Service.</p> <p>You will need to perform this step on the Meridium APM Application Server, the Meridium APM Policy Execution Server, <i>and</i> the Meridium APM Process Data Integration Server.</p>	<p>Always required on the Application Server and Policy Execution Server.</p> <p>Required on the Process Data Integration Server only if you want to use Xi Tag records in your policies.</p>
6	<p>Reset IIS.</p> <p>You will need to perform this step on the Meridium APM Application Server, the Meridium APM Policy Execution Server, <i>and</i> the Meridium APM Process Data Integration Server.</p>	<p>Always required on the Application Server and Policy Execution Server.</p> <p>Required on the Process Data Integration Server only if you want to use Xi Tag records in your policies.</p>
7	<p>Start (or restart if it is already started) the Meridium APM Process Data Integration Service on the Meridium APM Process Data Integration Server.</p>	<p>Required</p>
8	<p>Create a scheduled item in the Schedule Manager application in order for policies to be executed on a recurring basis.</p>	<p>Required</p>
9	<p>Assign the desired Security Users to one or more Policy Manager Security Groups via the Configuration Manager application.</p>	<p>Required</p>

Upgrade or Update Policy Manager to 3.6.1.1.0

The following tables list the steps that you must perform to update or upgrade Policy Manager to 3.6.1.1.0. These steps assume that you have upgraded the components in the basic Meridium APM system architecture.

Update from version V3.6.1.0.0

This module will be updated to 3.6.1.1.0 automatically when you update the components in the basic Meridium APM system architecture. No additional steps are required.

Upgrade from any version V3.6.0.0.0 through V3.6.0.12.3

Step	Task	Required/Optional	Notes
1	Upgrade the Meridium APM Policies Execution Service on the Meridium APM Policy Execution Server.	Required	None

Step	Task	Required/Optional	Notes
2	<p>Modify the Policies Execution Service configuration file to specify the Meridium APM Application Server, Meridium APM Execution Server, Meridium APM database, and login credentials.</p>	Required	<p>During the server upgrade process, a backup of your previously used configuration file was created. You can use this pre-upgrade file as a reference to modify the new baseline configuration file. You should <i>not</i> overwrite the baseline configuration file, as changes may have been made that are required for the service to work correctly. After you modify the baseline configuration file, you can delete the pre-upgrade configuration file.</p>
3	Start the Meridium APM Policies Execution Service.	Required	None

Step	Task	Required/Optional	Notes
4	<p>Modify the Policies Trigger Service configuration file to specify the Meridium APM Application Server, Meridium APM Execution Server, Meridium APM database, and login credentials.</p> <p>You will need to perform this step on the Meridium APM Application Server <i>and</i> the Meridium APM Process Data Integration Server.</p>	<p>Always required on the Application Server.</p> <p>Required on the Process Data Integration Server only if you want to use Xi Tag records in your policies.</p>	<p>During the server upgrade process, a backup of your previously used configuration file was created. You can use this pre-upgrade file as a reference to modify the new baseline configuration file. You should <i>not</i> overwrite the baseline configuration file, as changes may have been made that are required for the service to work correctly. After you modify the baseline configuration file, you can delete the pre-upgrade configuration file.</p>
5	<p>Start the Meridium APM Policies Trigger Service on the Meridium APM Application Server.</p> <p>You will need to perform this step on the Meridium APM Application Server <i>and</i> the Meridium APM Process Data Integration Server.</p>	<p>Always required on the Application Server.</p> <p>Required on the Process Data Integration Server only if you want to use Xi Tag records in your policies.</p>	None

Step	Task	Required/Optional	Notes
6	<p>Reset IIS.</p> <p>You will need to perform this step on the Meridium APM Application Server <i>and</i> the Meridium APM Process Data Integration Server.</p>	<p>Always required on the Application Server.</p> <p>Required on the Process Data Integration Server only if you want to use Xi Tag records in your policies.</p>	None
7	<p>Start (or restart if it is already started) the Meridium APM Process Data Integration Service on the Meridium APM Process Data Integration Server.</p>	Required	None
8	<p>Start the Scheduling Service on the Meridium APM Application Server.</p>	Required	None

Upgrade from any version V3.5.1.0.0 through V3.5.1.12.0

Step	Task	Required/Optional	Notes
1	<p>Upgrade the Meridium APM Policies Execution Service on the Meridium APM Policy Execution Server.</p>	Required	None

Step	Task	Required/Optional	Notes
2	<p>Modify the Policies Execution Service configuration file to specify the Meridium APM Application Server, Meridium APM Execution Server, Meridium APM database, and login credentials.</p>	Required	<p>During the server upgrade process, a backup of your previously used configuration file was created. You can use this pre-upgrade file as a reference to modify the new baseline configuration file. You should <i>not</i> overwrite the baseline configuration file, as changes may have been made that are required for the service to work correctly. After you modify the baseline configuration file, you can delete the pre-upgrade configuration file.</p>
3	Start the Meridium APM Policies Execution Service.	Required	None

Step	Task	Required/Optional	Notes
4	<p>Modify the Policies Trigger Service configuration file to specify the Meridium APM Application Server, Meridium APM Execution Server, Meridium APM database, and login credentials.</p> <p>You will need to perform this step on the Meridium APM Application Server <i>and</i> the Meridium APM Process Data Integration Server.</p>	<p>Always required on the Application Server.</p> <p>Required on the Process Data Integration Server only if you want to use Xi Tag records in your policies.</p>	<p>During the server upgrade process, a backup of your previously used configuration file was created. You can use this pre-upgrade file as a reference to modify the new baseline configuration file. You should <i>not</i> overwrite the baseline configuration file, as changes may have been made that are required for the service to work correctly. After you modify the baseline configuration file, you can delete the pre-upgrade configuration file.</p>
5	<p>Start the Meridium APM Policies Trigger Service on the Meridium APM Application Server.</p> <p>You will need to perform this step on the Meridium APM Application Server <i>and</i> the Meridium APM Process Data Integration Server.</p>	<p>Always required on the Application Server.</p> <p>Required on the Process Data Integration Server only if you want to use Xi Tag records in your policies.</p>	None

Step	Task	Required/Optional	Notes
6	<p>Reset IIS.</p> <p>You will need to perform this step on the Meridium APM Application Server <i>and</i> the Meridium APM Process Data Integration Server.</p>	<p>Always required on the Application Server.</p> <p>Required on the Process Data Integration Server only if you want to use Xi Tag records in your policies.</p>	None
7	<p>Start (or restart if it is already started) the Meridium APM Process Data Integration Service on the Meridium APM Process Data Integration Server.</p>	Required	None
8	<p>Start the Scheduling Service on the Meridium APM Application Server.</p>	Required	None

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

Step	Task	Required/Optional	Notes
1	<p>Upgrade the Meridium APM Policies Execution Service on the Meridium APM Policy Execution Server.</p>	Required	None

Step	Task	Required/Optional	Notes
2	<p>Modify the Policies Execution Service configuration file to specify the Meridium APM Application Server, Meridium APM Execution Server, Meridium APM database, and login credentials.</p>	Required	<p>During the server upgrade process, a backup of your previously used configuration file was created. You can use this pre-upgrade file as a reference to modify the new baseline configuration file. You should <i>not</i> overwrite the baseline configuration file, as changes may have been made that are required for the service to work correctly. After you modify the baseline configuration file, you can delete the pre-upgrade configuration file.</p>
3	Start the Meridium APM Policies Execution Service.	Required	None

Step	Task	Required/Optional	Notes
4	<p>Modify the Policies Trigger Service configuration file to specify the Meridium APM Application Server, Meridium APM Execution Server, Meridium APM database, and login credentials.</p> <p>You will need to perform this step on the Meridium APM Application Server <i>and</i> the Meridium APM Process Data Integration Server.</p>	<p>Always required on the Application Server.</p> <p>Required on the Process Data Integration Server only if you want to use Xi Tag records in your policies.</p>	<p>During the server upgrade process, a backup of your previously used configuration file was created. You can use this pre-upgrade file as a reference to modify the new baseline configuration file. You should <i>not</i> overwrite the baseline configuration file, as changes may have been made that are required for the service to work correctly. After you modify the baseline configuration file, you can delete the pre-upgrade configuration file.</p>
5	<p>Start the Meridium APM Policies Trigger Service on the Meridium APM Application Server.</p> <p>You will need to perform this step on the Meridium APM Application Server <i>and</i> the Meridium APM Process Data Integration Server.</p>	<p>Always required on the Application Server.</p> <p>Required on the Process Data Integration Server only if you want to use Xi Tag records in your policies.</p>	None

Step	Task	Required/Optional	Notes
6	<p>Reset IIS.</p> <p>You will need to perform this step on the Meridium APM Application Server <i>and</i> the Meridium APM Process Data Integration Server.</p>	<p>Always required on the Application Server.</p> <p>Required on the Process Data Integration Server only if you want to use Xi Tag records in your policies.</p>	None
7	Start (or restart if it is already started) the Meridium APM Process Data Integration Service on the Meridium APM Process Data Integration Server.	Required	None
8	Start the Scheduling Service on the Meridium APM Application Server.	Required	None

Upgrade from any version V3.5.0 through V3.5.0.0.7.1

Step	Task	Required/Optional	Notes
1	Upgrade the Meridium APM Policies Execution Service on the Meridium APM Policy Execution Server.	Required	None

Step	Task	Required/Optional	Notes
2	<p>Modify the Policies Execution Service configuration file to specify the Meridium APM Application Server, Meridium APM Execution Server, Meridium APM database, and login credentials.</p>	Required	<p>During the server upgrade process, a backup of your previously used configuration file was created. You can use this pre-upgrade file as a reference to modify the new baseline configuration file. You should <i>not</i> overwrite the baseline configuration file, as changes may have been made that are required for the service to work correctly. After you modify the baseline configuration file, you can delete the pre-upgrade configuration file.</p>
3	Start the Meridium APM Policies Execution Service.	Required	None

Step	Task	Required/Optional	Notes
4	<p>Modify the Policies Trigger Service configuration file to specify the Meridium APM Application Server, Meridium APM Execution Server, Meridium APM database, and login credentials.</p> <p>You will need to perform this step on the Meridium APM Application Server <i>and</i> the Meridium APM Process Data Integration Server.</p>	<p>Always required on the Application Server.</p> <p>Required on the Process Data Integration Server only if you want to use Xi Tag records in your policies.</p>	<p>During the server upgrade process, a backup of your previously used configuration file was created. You can use this pre-upgrade file as a reference to modify the new baseline configuration file. You should <i>not</i> overwrite the baseline configuration file, as changes may have been made that are required for the service to work correctly. After you modify the baseline configuration file, you can delete the pre-upgrade configuration file.</p>
5	<p>Start the Meridium APM Policies Trigger Service on the Meridium APM Application Server.</p> <p>You will need to perform this step on the Meridium APM Application Server <i>and</i> the Meridium APM Process Data Integration Server.</p>	<p>Always required on the Application Server.</p> <p>Required on the Process Data Integration Server only if you want to use Xi Tag records in your policies.</p>	None

Step	Task	Required/Optional	Notes
6	<p>Reset IIS.</p> <p>You will need to perform this step on the Meridium APM Application Server <i>and</i> the Meridium APM Process Data Integration Server.</p>	<p>Always required on the Application Server.</p> <p>Required on the Process Data Integration Server only if you want to use Xi Tag records in your policies.</p>	None
7	Start (or restart if it is already started) the Meridium APM Process Data Integration Service on the Meridium APM Process Data Integration Server.	Required	None
8	Start the Scheduling Service on the Meridium APM Application Server.	Required	None

How are Policies Executed?

When a policy is executed, records that meet the conditions defined in the policy logic are used to trigger actions that are defined by the Action nodes in the policy diagram.

You can configure a policy to be executed:

- [When records or tag values associated with the policy are updated.](#)
 - If the policy includes records from the Meridium APM database, the policy will be executed when those records are updated.
 - If the policy includes an Xi Tag record, the policy will be executed when the reading value for that tag changes in the process historian or when the Xi Tag record in the Meridium APM database is updated.
- [On a predefined schedule.](#)

About Configuring Policies to be Executed When Records or Tag Values are Updated

If you configure a policy to be executed when records or tag values in the policy are updated, the following items are used in the execution process:

- **Meridium APM Policies Execution Service:** A Windows service that is installed automatically on the Meridium APM Policy Execution Server when you select the **Meridium APM Policy Execution Service** option in the Meridium APM Server and Add-ons installer.

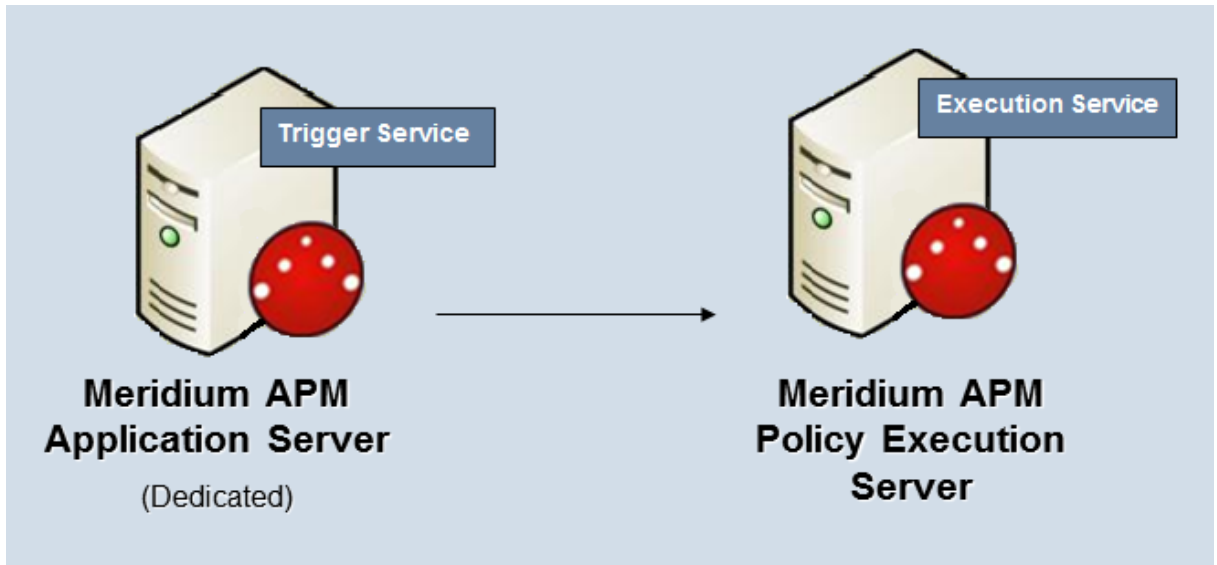
Note: Throughout the rest of this documentation, we refer to this service as the *Execution Service*.

- **Meridium APM Policies Trigger Service:** A Windows service that is installed automatically on:
 - The *Meridium APM Application Server* when you select the **Meridium APM Application Server** option in the Meridium APM Server and Add-ons installer.
 - The *Meridium APM Process Data Integration Server* when you select the **Meridium APM Process Data Integration Service** option in the Meridium APM Server and Add-ons installer.

Note: Throughout the rest of this documentation, we refer to this service as the *Trigger Service*.

The Execution Service and the Trigger Service work together to facilitate the policy execution process. An *Execution Service* is *always* used. At least *one* of the Trigger Services is also always used. Depending upon the type of records that are included in the policy, either the one on the Meridium APM Application Server or the one on the Meridium APM Process Data Integration Server is used.

- The Trigger Service on the Meridium APM *Application Server* is used when records are updated. The following image illustrates the services that are used in this scenario.

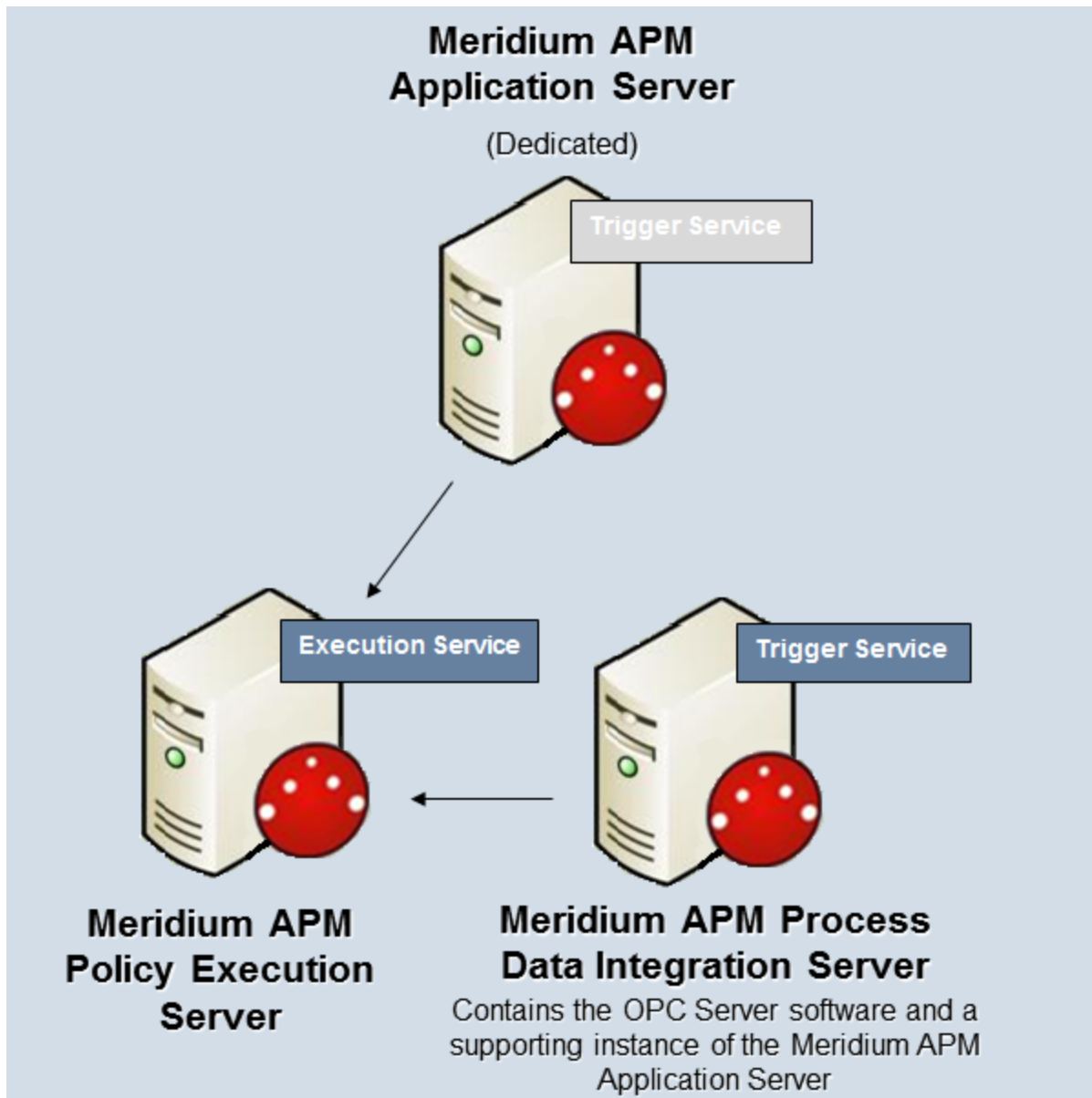


This image depicts one Policy Execution Server. Depending upon the number of policies that you will need to manage in your system, you may have multiple Meridium APM Policy Execution Server machines.

- The Trigger Service on the Meridium APM *Process Data Integration Server* is used when a policy includes an Xi Tag record whose reading value changes in the process historian.

Note: Your system architecture will include a Meridium APM Process Data Integration Server only if you are using the Meridium APM Process Data Integration feature.

The following image illustrates the services that are used in this scenario. Note that the Trigger Service on the Meridium APM Application Server exists but is shaded gray to indicate that it is not used in this scenario.



About Configuring Policies to be Executed on a Predefined Schedule

If you configure a policy to be executed on a predefined schedule, the following items are used in the execution process:

- **The master scheduled item.** Checks the policy execution schedule in all *active* policies and executes the policies that are due.
- **The policy execution schedule.** Determines how often the policy should be executed.

Note: Each policy has its own policy execution schedule. This means that while there will be multiple policy execution schedules, you need to configure only *one* master schedule.

The policy execution schedule should be *less* frequent than the schedule defined in the scheduled item. Otherwise, the policy will be executed only when the scheduled item runs.

For example, suppose the following schedules have been configured:

- **Scheduled Item Schedule:** Every day
- **Policy Execution Schedule:** Every hour

In this case, although the policy execution schedule indicates that the policy should be executed every *hour*, the policy will actually be executed every *day* when the scheduled item runs. In order for the policy to be executed every hour, the scheduled item must be run *more often* than every hour (e.g., every fifteen minutes).

For this reason, we recommend that you configure the scheduled item to be run every *15 minutes*.

Installing the Meridium APM Policies Execution Service

The Meridium APM Policies Execution Service must be installed on the machine that will serve as the Meridium APM Policy Execution Server. We recommend that this is a separate machine from the dedicated Meridium APM Application Server. The following instructions provide details on installing the Meridium APM Policies Execution Service using the Meridium APM Server and Add-ons installer.

These instructions assume that:

- You are installing the Meridium APM Policies Execution Service, along with a supporting instance of the Meridium APM Application Server software, on a separate machine from the dedicated Meridium APM Application Server.
- Your system meets the hardware and software requirements for the basic Meridium APM system architecture.

Note: If your system does not meet certain software requirements, a message may be displayed during the installation process that indicates the software requirement that is missing. Some software requirements, however, will be installed automatically if they are missing.

- The [Meridium APM Policy Execution Server machine meets the hardware and software requirements](#).
- You are an Administrator with full access to the Application Server machine.

Note: IIS will be reset automatically by the installer before the installation process begins.

To install the Meridium APM Policy Execution Service software on the Policy Execution Server machine:

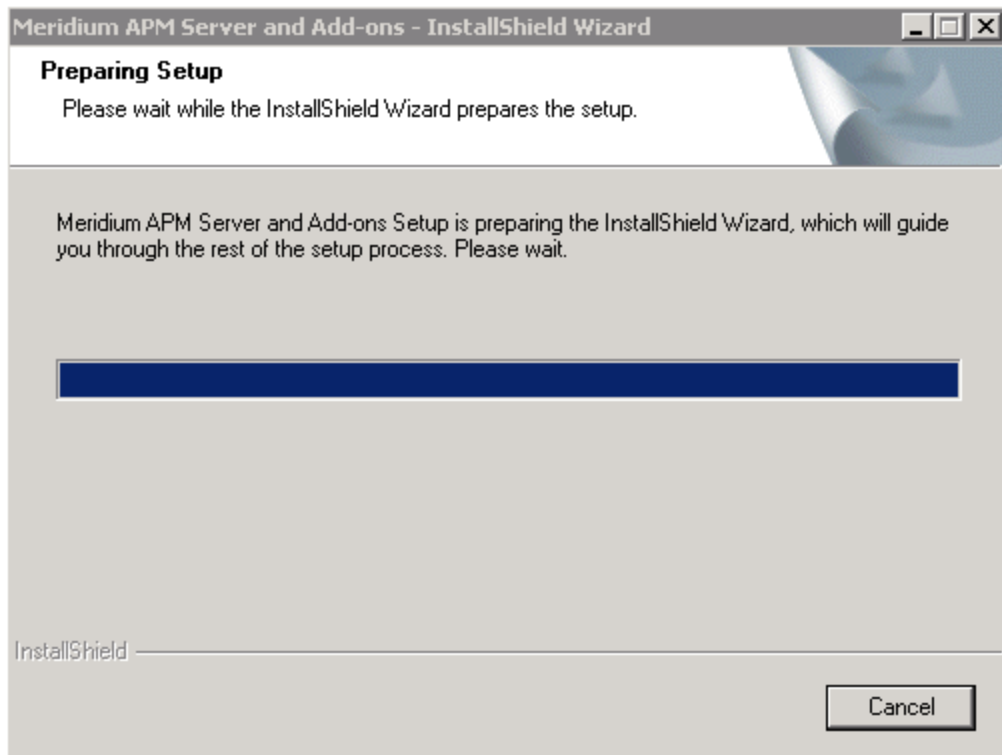
1. On the Meridium APM Policy Execution Server machine, insert the Meridium APM distribution DVD, and navigate to the folder **General Release\Meridium APM Framework and Server Setup\Setup\Meridium APM Server and Add-ons**.

2. Double-click the file **Setup.exe**.

A message appears, asking if you want to allow the setup.exe to make changes to your machine.

3. Click the **Yes** button.

The **Meridium APM Server and Add-ons** installer appears, displaying the **Preparing Setup** screen, which contains a progress bar.

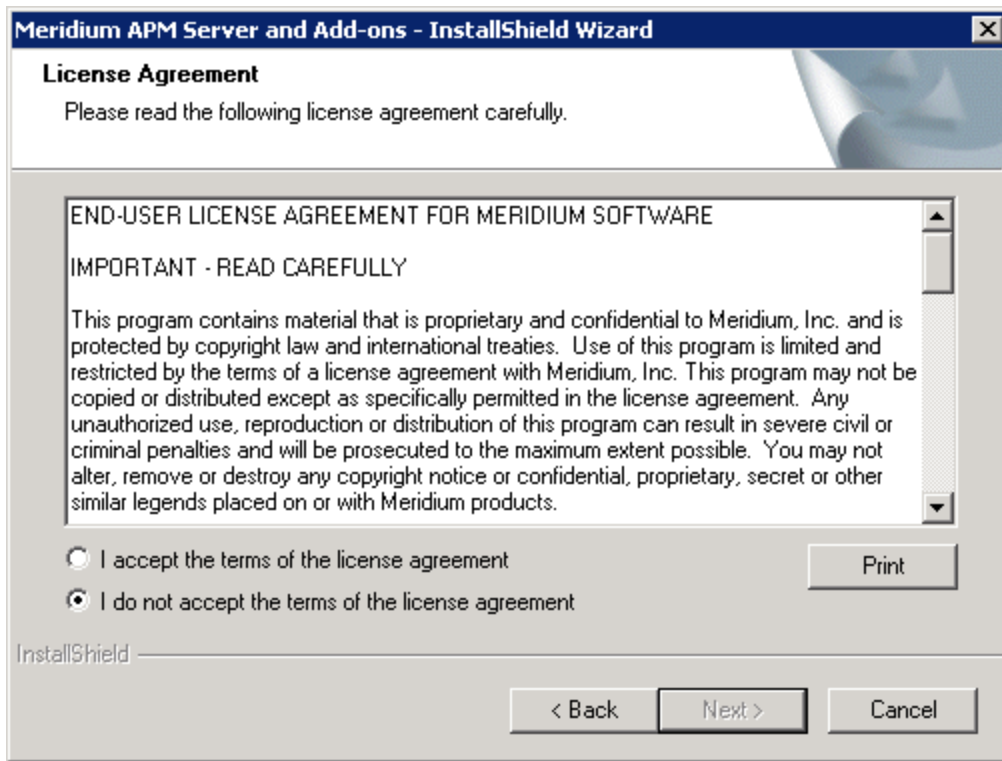


After the progress bar reaches the end, the Welcome screen appears.



4. Click the **Next** button.

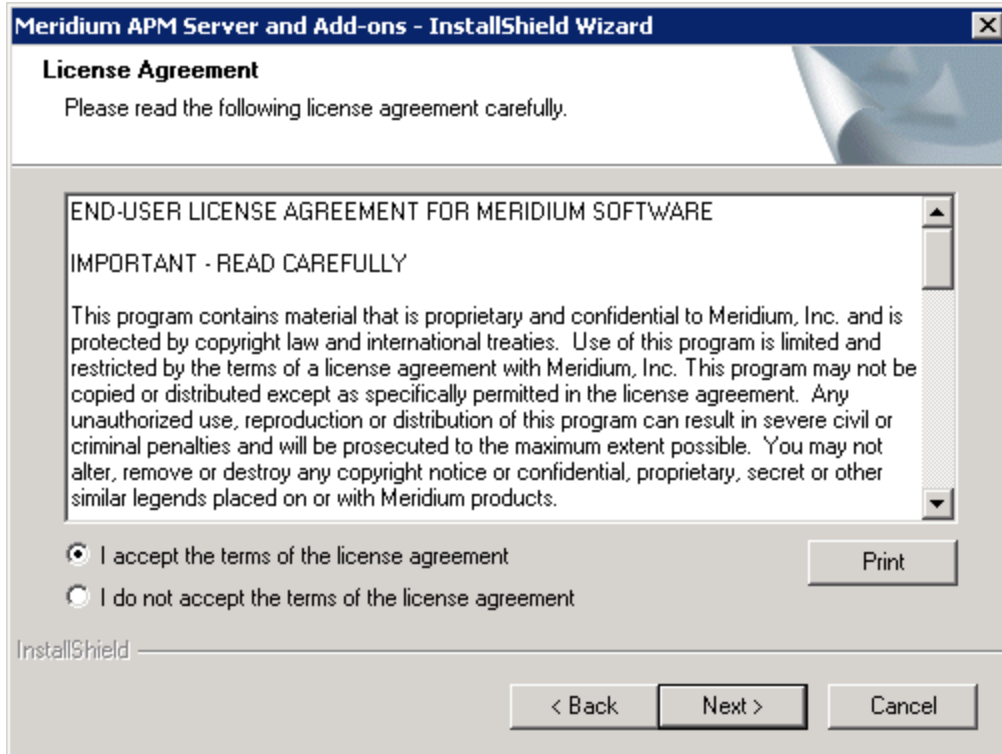
The **License Agreement** screen appears.



5. Read the entire license agreement, and then select one of the following options:

- **I accept the terms of the license agreement**, if you agree to the terms of the license agreement and want to continue. These instructions assume that you want to continue.
- **I do not accept the terms of the license agreement**. This option is selected by default. If you do not agree to the terms of the license agreement and do not want to continue, click the **Cancel** button to exit the installer.

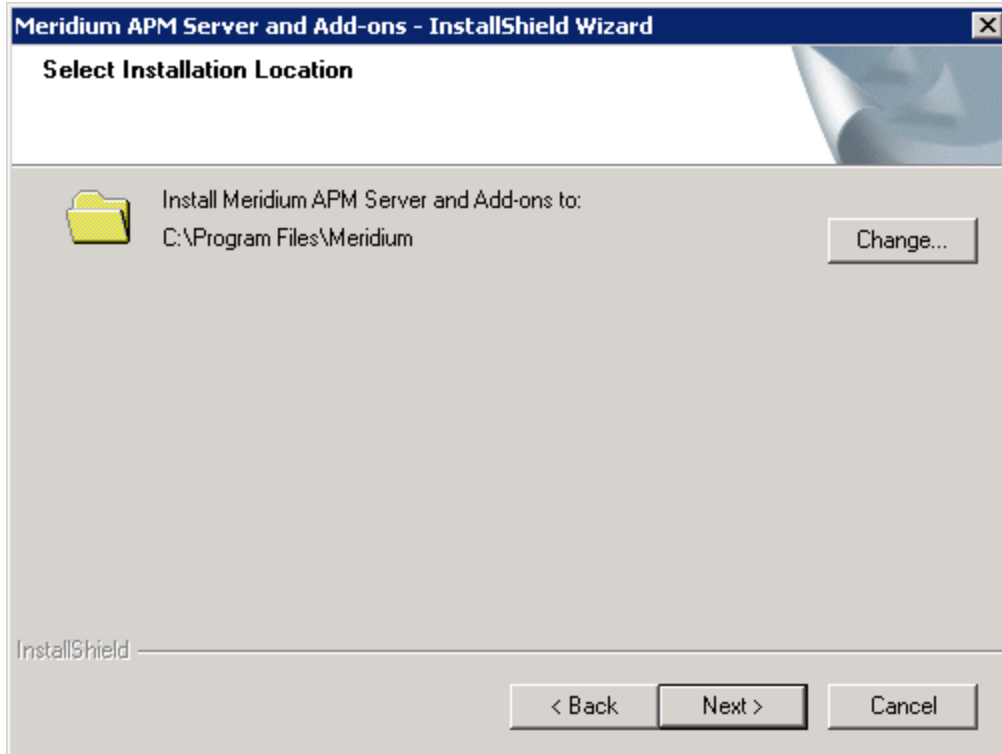
The **Next** button is enabled.



OE Hint: If you want to print the license agreement, click the **Print** button.

6. Click the **Next** button.

The **Select Installation Location** screen appears, prompting you to select the location where the Policy Execution Service software will be installed. The software will be saved to the following folder on the Meridium APM Policy Execution Server by default: **C:\Program Files\Meridium**.

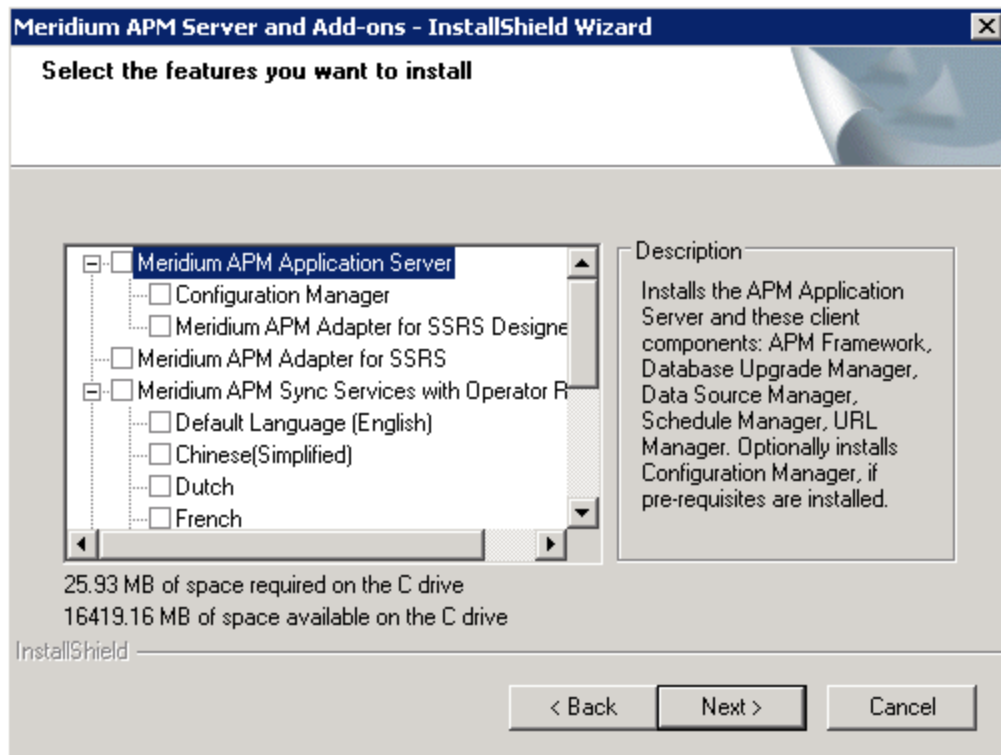


7. If you are satisfied with the default location where the software will be installed, click the **Next** button.

-or-

If you want to change the location where the software will be installed, click the **Change** button, and navigate to the location where you want to install the software. The folder path that you select will be displayed in place of the default folder path. When you are satisfied with the installation location, click the **Next** button.

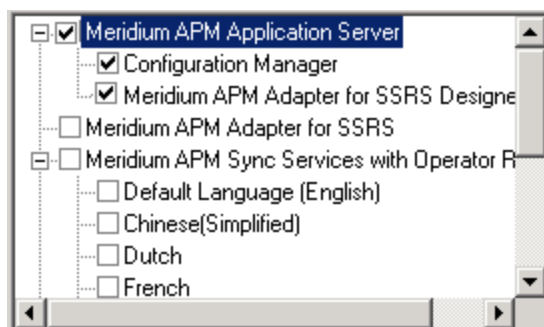
The **Select the features you want to install** screen appears, and the **Meridium APM Application Server** node is highlighted automatically in the tree.



IMPORTANT: The **Select the features you want to install** screen lets you specify which features you want to install on the Policy Execution Server machine. *These instructions* assume that you want to deploy only the Meridium APM Application Server and the Meridium APM Policy Execution Service software. When the Policy Execution Service software is installed, the Meridium APM System Administration Tool will be installed automatically. When the Meridium APM Application Server software is installed, the following Meridium APM applications are installed automatically: APM Framework, Database Upgrade Manager, Data Source Manager, Schedule Manager, and URL Manager. Corresponding options for these applications are not listed in the tree because these components must be installed with the Meridium APM Application Server software. You can optionally install the Configuration Manager application and Meridium APM Adapter for SSRS Designer.

8. Select the **Meridium APM Application Server** check box.

The successor nodes are selected automatically.



9. Clear the **Configuration Manager** check box.
10. Clear the **Meridium APM Adapter for SSRS Designer** option.
11. Select the **Meridium APM Policy Execution Service** option.

IMPORTANT: While additional options are available for selection in the tree on the **Select the features you want to install** screen, these options are not meant to be installed on the Policy Execution Server. *These instructions* assume that you want to install the Meridium APM Application Server software (with APM Framework applications) and the Policy Execution Service software. This installer is also used to install the additional components *on other machines*.

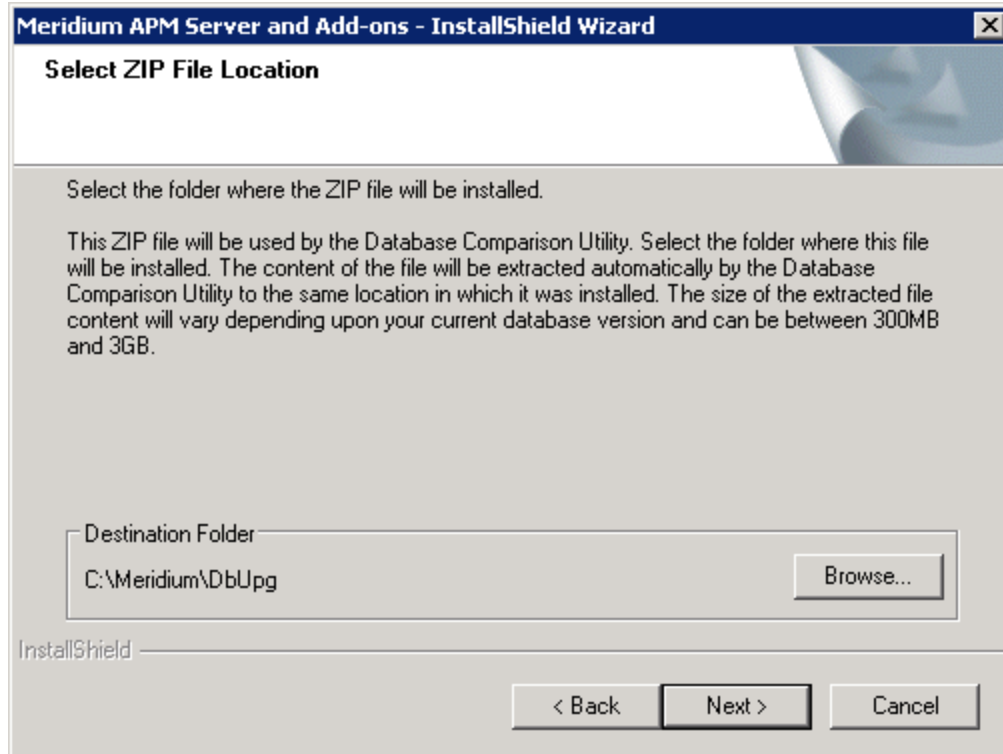
12. Click the **Next** button.

A message appears, indicating that the installer is checking your machine for the required prerequisites for the features that you want to install.

- If one or more prerequisites are missing on the machine, the **Meridium Installer** screen will appear and display a message that indicates which prerequisites still need to be installed on the machine before you can install the feature that is dependent on that prerequisite. This message also indicates what you can do to continue.

If you see this screen, you should read the message in detail, and either click the **Back** button to clear the selection whose prerequisites are missing, and then continue through the installation without installing that component, or close the installer, install the missing prerequisite, and then run the installer again later.

- If all the prerequisites for the selected components are installed on the machine, or you have selected components that do not require any prerequisites, the **Select ZIP File Location** screen appears.



The **Select ZIP File Location** screen lets you specify the location where you want the ZIP file of database content to be installed. This ZIP file is used by the Database Comparison Utility. By default, the ZIP file will be installed in the folder **C:\Meridium\DbUpg**. You can, however, select a different location.

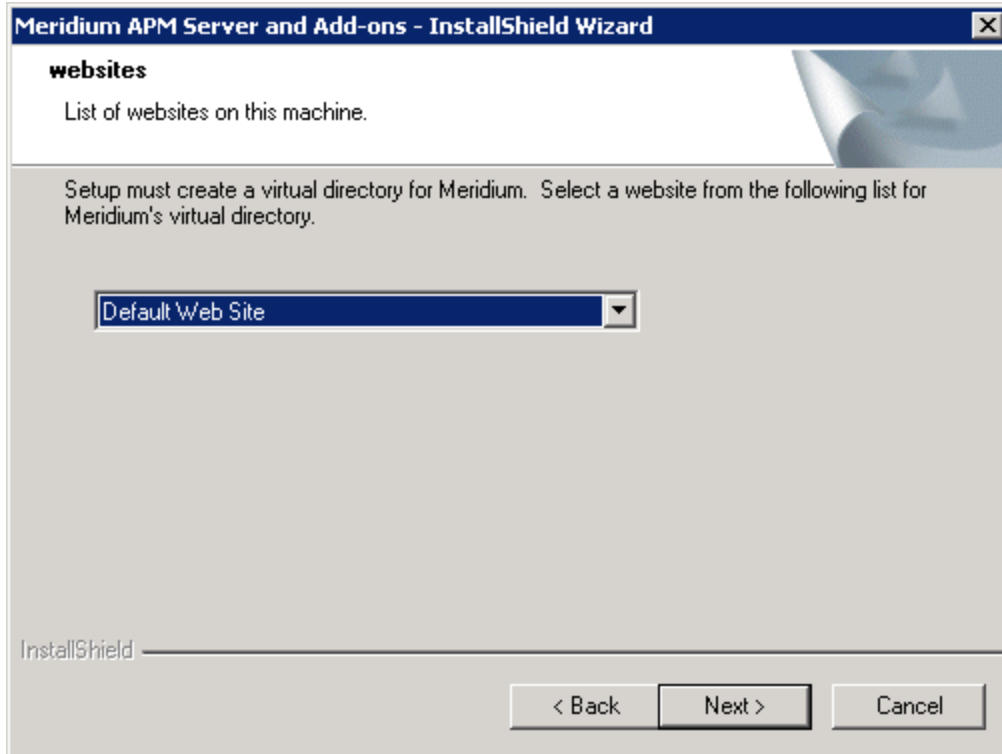
13. Read the information on this screen and note the estimated size of the extracted files from this ZIP file. You should use this information to determine where you want install the ZIP file.
14. If you want the ZIP file to be installed in the default location, accept the default setting, and move on the next step.

-or-

If you want the ZIP file to be installed in a different location, click the **Browse** button, and navigate to the location where you want to ZIP file to be installed. The selected file path will appear in the **Destination Folder** section in place of the default file path.

15. Click the **Next** button.

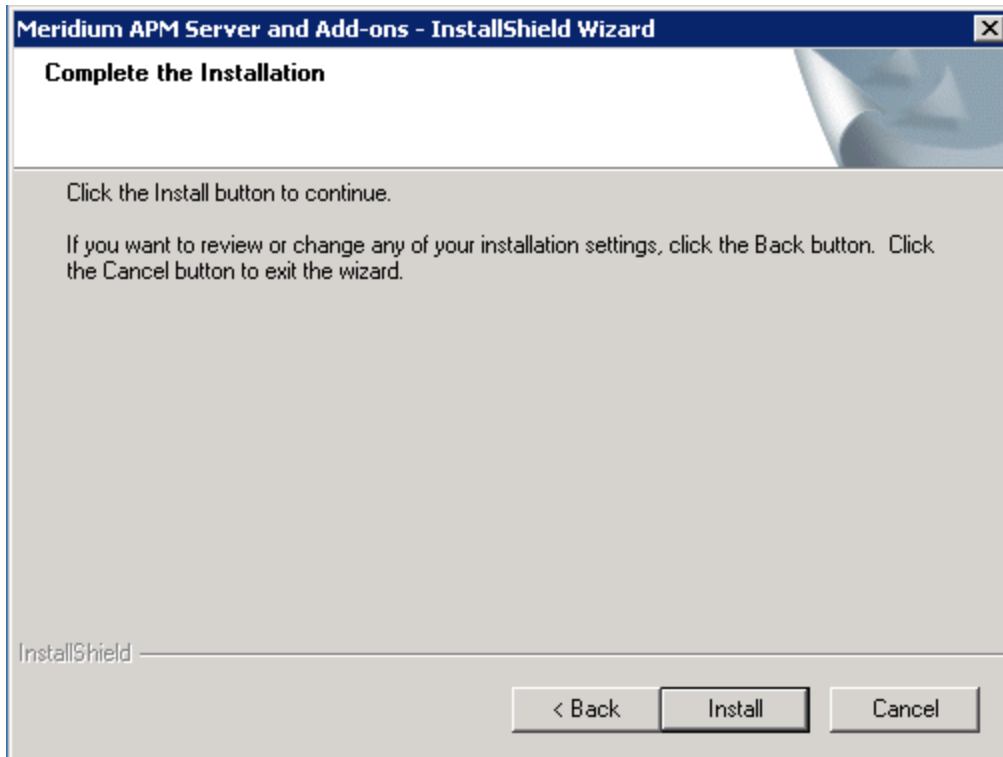
The **websites** screen appears.



The **websites** screen lets you specify where the installer will create a virtual directory for Meridium APM. This website is configured in the IIS Manager on the APM Application Server machine. The websites list contains all the websites that are configured on the APM Application Server machine. The default website is *Default Web Site*. Throughout this documentation, we assume that you have chosen to install Meridium APM under the *Default Web Site*.

16. In the list of websites, select the website where you want the installer to create a virtual directory.
17. Click the **Next** button.

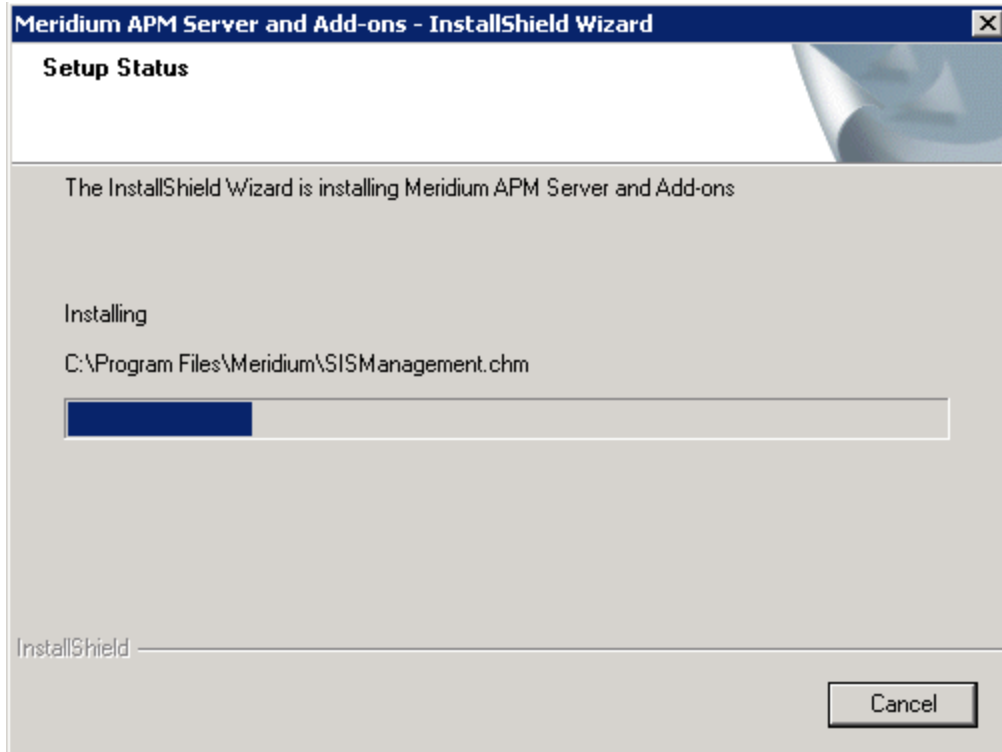
The **Complete the Installation** screen appears.



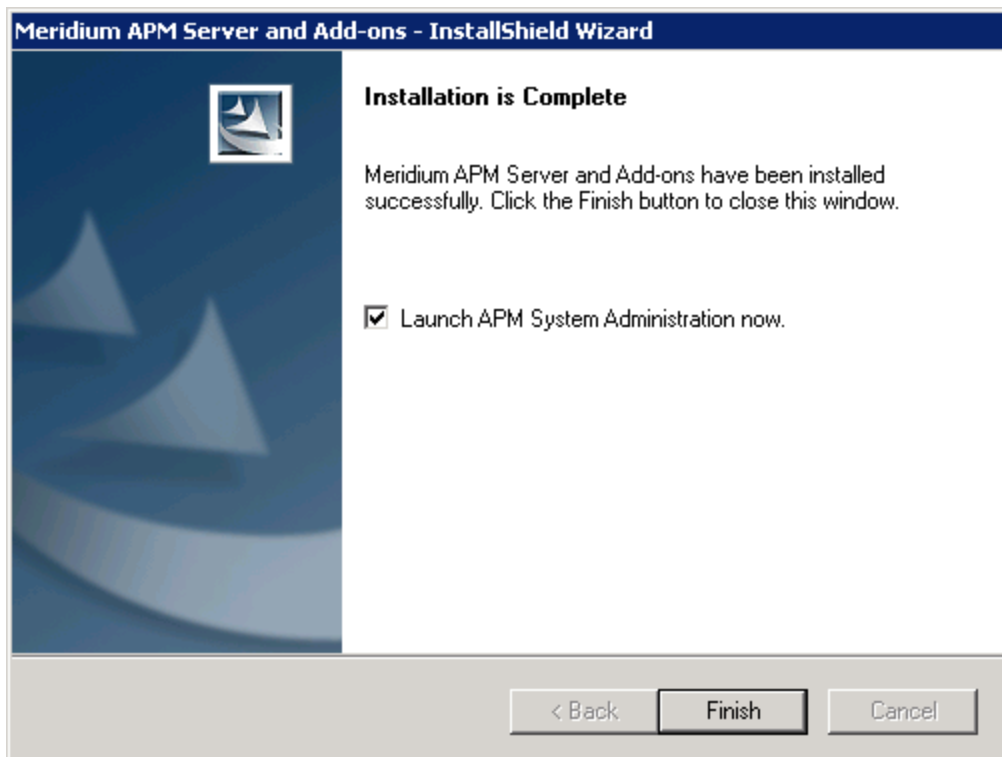
The **Complete the Installation** screen contains a message, indicating that if you want to modify any of the settings that you specified in the preceding screens in the installer, you can use the **Back** button to do so, or if you do not want to complete the installation, you can click **Cancel**. These instructions assume that you are ready to complete the installation process.

18. Click the **Install** button.

The **Setup Status** screen appears, which displays a progress bar that shows the progress of the installation process.



After the progress bar reaches the end, a message appears, indicating that your server is being configured. After your server is configured, the **Installation is Complete** screen appears.



This screen contains a message, indicating that the software has been installed successfully. You can also select to optionally launch the APM System Administration tool when the installer window closes. By default, this option is set to Yes.

19. If you want the APM System Administration tool to be launched as soon as you click the **Finish** button, accept the default selection.

-or-

If you do not want the APM System Administration tool to be launched at this time, clear the **Launch APM System Administration now** check box.

20. Click the **Finish** button.

The **Meridium APM Server and Add-ons** installer closes. Additionally, if the **Launch APM System Administration now** check box was selected, the **APM System Administration** window appears. At this point, you should [refer back to the checklist](#).

Modifying the File Meridium.Policies.ExecutionService.exe.config

The following instructions provide details on modifying the file Meridium.Policies.ExecutionService.exe.config manually.

To modify the file Meridium.Policies.ExecutionService.exe.config:

1. On the Meridium APM Policy Execution Server, navigate to the folder where the Policies Execution Service files are installed. If you installed the Meridium Policy Execution Services software in the default location, you can locate this file in the following location: **C:\Program Files\Meridium\Services**
2. Open the file **Meridium.Policies.ExecutionService.exe.config** in an application that you can use to modify XML code.
3. In the file, within the **<baseAddresses>** tags, locate the following text:

```
<add baseAddress="http://localhost:8000/Meridium/Policies/ExecutionService"/>
```

4. Replace **localhost** with the name or IP address of the Meridium APM Policy Execution Server where the Policies Execution Service is installed.

Note: If you do not modify this value, your policies will not be updated.

5. Within the **<meridiumConnections>** tags, uncomment the example connection tag by deleting **<!--EXAMPLE:** and the corresponding **-->** from the beginning and end of the string.
6. Within the **<meridiumConnections>** tags, within the **connection name** attribute, replace **TEST** with a name for the connection to the database. This value is used only by the configuration file.
7. Within the **<meridiumConnections>** tags, within the **applicationServer** attribute:
 - If you want the Meridium APM Policy Execution Server to use a remote connection to the Meridium APM Application Server, between the quotations, type the name of the Meridium APM Application Server.
 - or-
 - If you want the Meridium APM Policy Execution Server to use a local connection to the Meridium APM Application Server, accept the default setting (i.e., the empty string).
8. Within the **<meridiumConnections>** tags, within the text **datasource** attribute, replace **DVT_ALL_350_2** with the name of the Meridium APM database that you will use to store your policies.

Note:The Datasource value is case sensitive. It is very important that you type the database name exactly as it is defined in the Datasource Manager. Keep in mind that the datasource that you specify here must also match the datasource that you specify for any other database setting on other machines, or you policies will not be triggered as expected.

9. Within the **<meridiumConnections>** tags, within the **userId** attribute, replace **MIADMIN** with the user name of the user whose credentials should be used to log in to the Meridium APM database specified as the **datasource** value.

Note:We recommend that the user you specify here is a Super User. Otherwise, the user that you specify must have privileges to all Policy Manager families and any other families that will be associated with a policy.

10. Within the **<meridiumConnections>** tags, within the **password** attribute, replace **PaSsWoRd** with the password that is associated with the user name specified as the **userId** value. Note that you should not delete the **!** in front of the password. This symbol is not part of the password itself. Instead, this symbol will cause the password to be encrypted automatically when the service is restarted.
11. Save the file.

Your settings will be applied when you start or restart the Policy Execution Service.

Modifying the File Meridium.Policies.Service.exe.config

The following instructions provide details on modifying the file Meridium.Policies.Service.exe.config manually on the Meridium APM Application Server, the Meridium APM Policy Execution Server, or the Meridium APM Process Data Integration Server.

To modify the Meridium.Policies.Service.exe.config file:

1. On the machine on which you are configuring the Meridium Policies Trigger Service, navigate to the folder where the Policies Execution Service files are installed. If you installed the Production Data Integration software in the default location, you can locate this file in the following location: C:\Program Files\Meridium\Services.
2. Open the file *Meridium.Policies.Service.exe.config* in an application that you can use to modify XML script (e.g., Notepad).
3. In the file, within the **<ExecutionServers>** tags, uncomment the example connection tag by deleting **<!--EXAMPLE:** and the corresponding **-->** from the beginning and end of the string.
4. Within the **<Execution Servers>** tags, within the **add url** attribute:
 - If you are modifying this file on the *Meridium APM Application Server* or the *Meridium APM Process Data Integration Server*, replace **localhost** with the name or IP address of the Policy Execution Server where the Policy Execution Service is installed.

-or-

 - If you are modifying this file on the *Meridium APM Policy Execution Server*, accept the default value (i.e., *localhost*).
3. In the file, within the **<meridiumConnections>** tags, uncomment the example connection tag by deleting **<!--EXAMPLE:** and the corresponding **-->** from the beginning and end of the string.
6. Within the **<meridiumConnections>** tags, within the **connection name** attribute, replace **TEST** with a name for the connection to the database. This value is used only by the configuration file.
7. Within the **<meridiumConnections>** tags, within the **applicationServer** attribute:
 - If you want the triggering service to use a remote connection to the Application Server, between the quotations, type the Application Server name.

-or-

 - If you want the triggering service to use a local connection to the Application Server, accept the default setting, and move on to the next step.

8. Within the **<meridiumConnections>** tags, within the **datasource** attribute, replace **DVT_ALL_350_2** with the name of the Meridium APM database that you will use to store your policies.

Note: The Datasource value is case sensitive. It is very important that you type the database name exactly as it is defined in the Datasource Manager. Keep in mind that the datasource that you specify here must also match the datasource that you specify for any other database setting on other machines, or you policies will not be triggered as expected.

9. Within the **<meridiumConnections>** tags, within the **userId** attribute, replace **MIADMIN** with the user name of the user whose credentials should be used to log in to the Meridium APM database specified as the **datasource** value.

Note: We recommend that the user you specify here is a Super User. Otherwise, the user that you specify must have privileges to all Policy Manager families and any other families that will be associated with a policy.

10. Within the **<meridiumConnections>** tags, within the **password** attribute, replace **PaSsWoRd** with the password that is associated with the user name specified as the **userId** value. Note that you should not delete the **!** in front of the password. This symbol is not part of the password itself. Instead, this symbol will cause the password to be encrypted automatically when the service is restarted.
11. Save the file.

Your settings will be applied when you start or restart the Policies Trigger Service.

Creating the Scheduled Item to Execute Policies on a Predefined Schedule

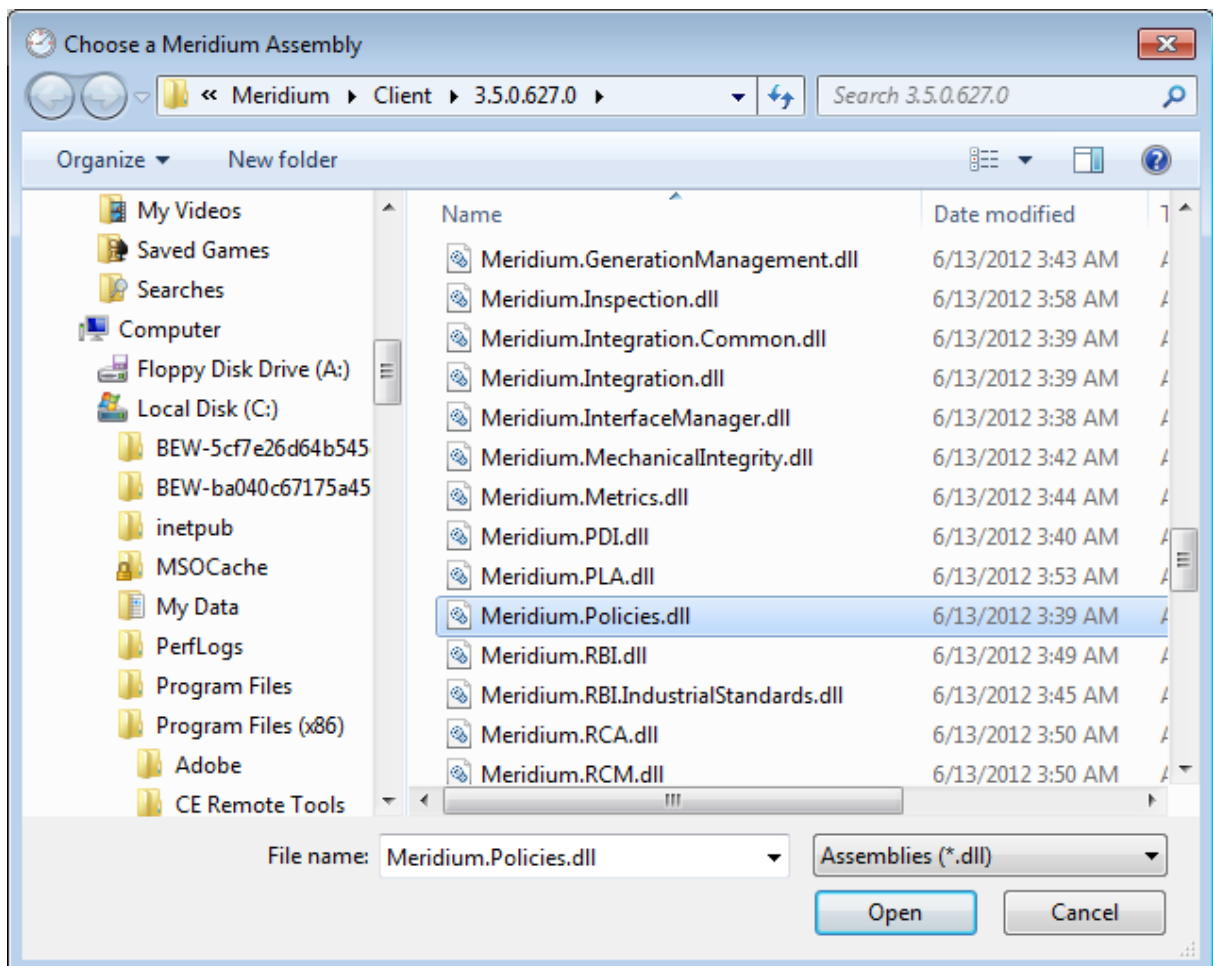
Before you can [schedule individual policies to be executed on a recurring basis](#), you must first set up a scheduled item in Meridium APM Schedule Manager.

Note: You must be a Super User or a member of the MI Policy Designer Security Group in order to create the scheduled item for Policy Manager.

To create the scheduled item for policies:

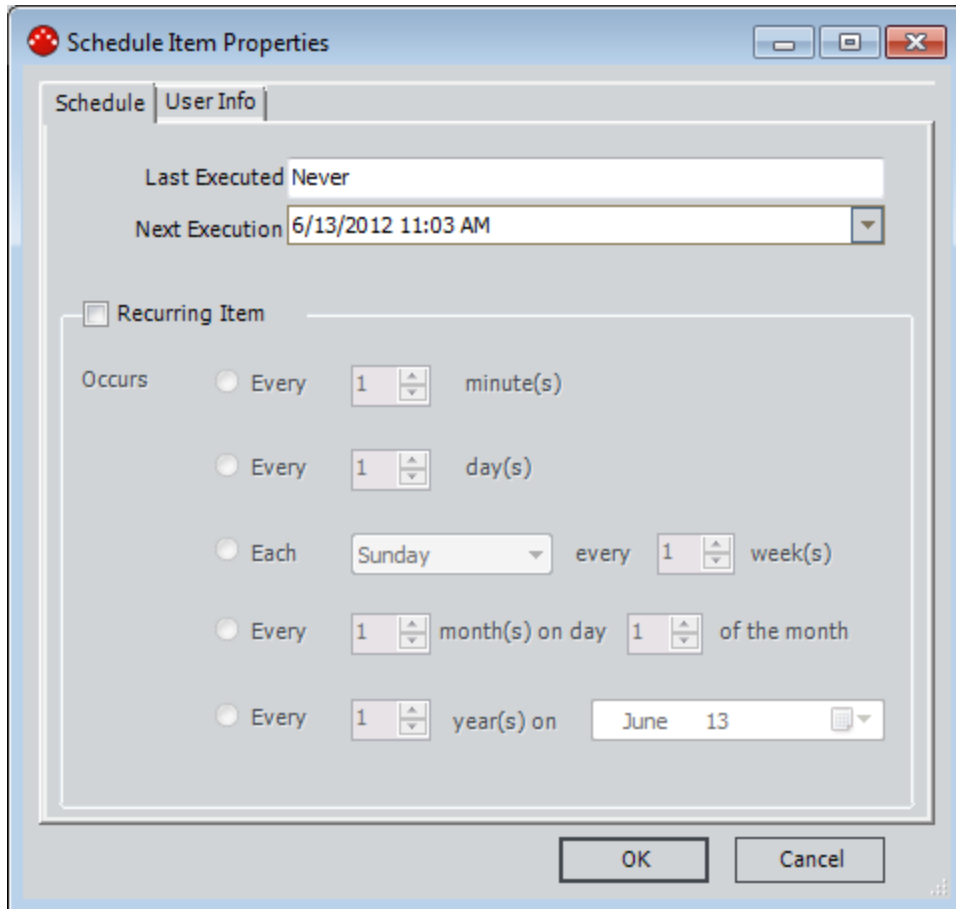
1. Launch the Meridium APM Schedule Manager and log in.
2. On the **Scheduled Items** window, click the **Add New** link.

The **Choose a Meridium Assembly** window appears.



3. Select the file **Meridium.Policies.dll**, and click the **Open** button.

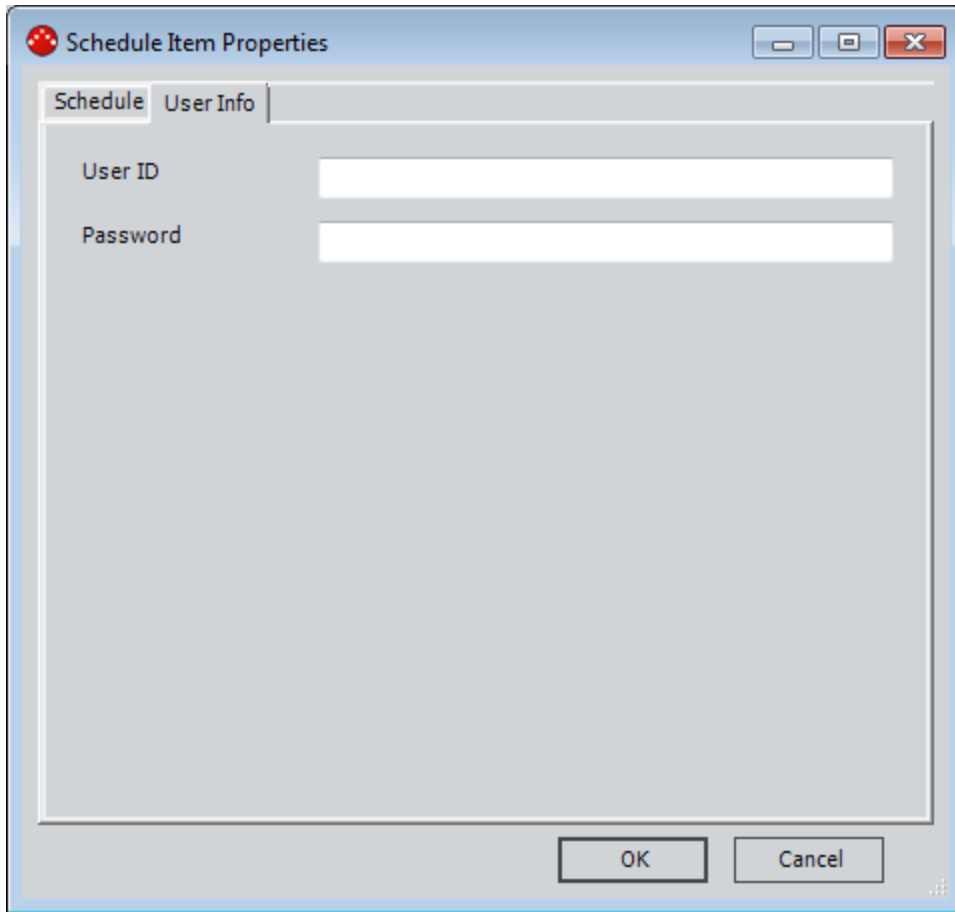
The **Schedule Item Properties** dialog box appears, displaying the **Schedule** tab.



4. Select the **Recurring Item** check box, and then select the frequency for the execution.

Note: The recommended frequency for the scheduled item is 15 minutes.

5. Click the **User** tab.



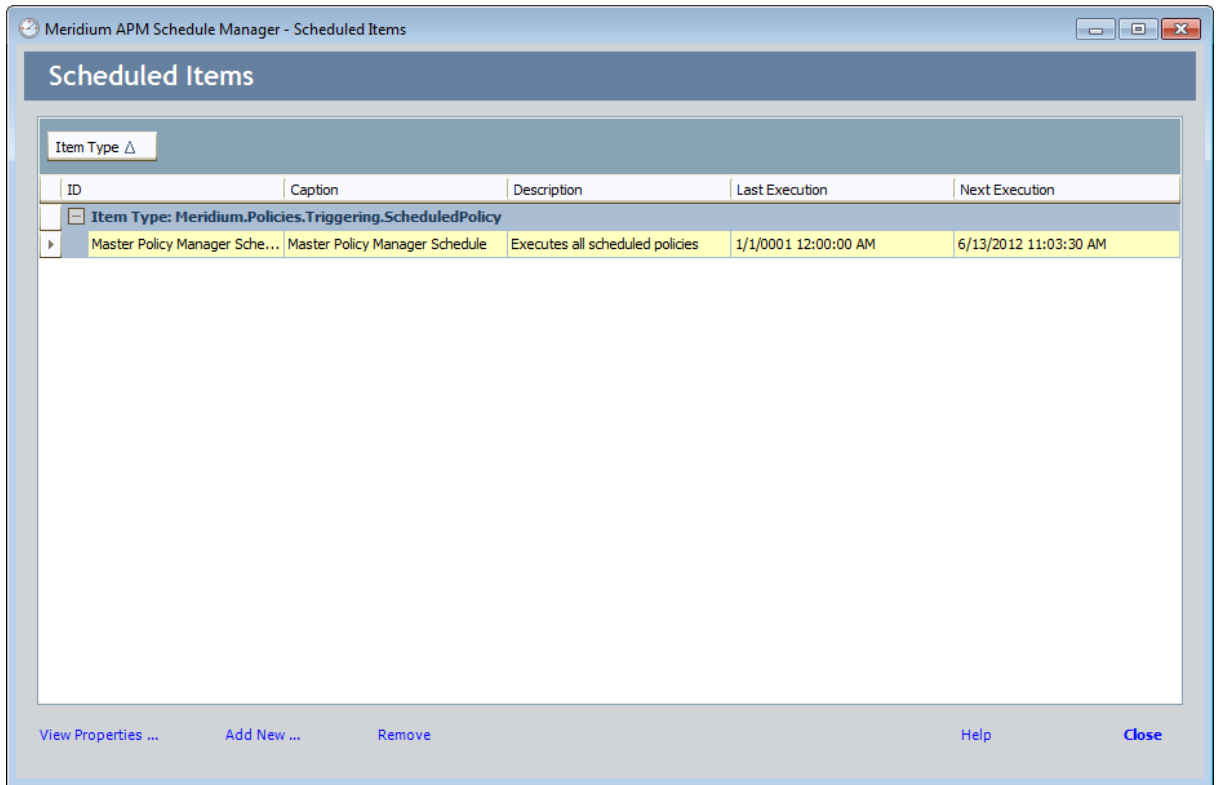
6. In the **User ID** text box, type the User ID of the Security User whose account you want to use for executing this scheduled item.

Note: If you specify the User ID of a Security User who is not a Super User or a member of the MI Policy Designer Security Group, when you click **OK**, a message will appear stating that you have specified an invalid user, and the item will not be scheduled.

7. In the **Password** text box, type the password associated with the User ID that you specified in the previous step.
8. Click **OK**.

The scheduled item is saved and appears in the **Scheduled Items** window.

Creating the Scheduled Item to Execute Policies on a Predefined Schedule



Policy Manager Security Groups

The following Security Groups are provided for use with Policy Manager:

- MI Policy Designer
- MI Policy User
- MI Policy Viewer

The following table lists the default privileges that members of each group have to the Policy Manager families.

Family	MI Policy Designer	MI Policy User	MI Policy Viewer
Health Indicator Value	View, Update, Insert, Delete	None	View
Policy	View, Update, Insert, Delete	View	View
Policy Instance	View, Update, Insert, Delete	View, Update, Insert, Delete	View
Policy Recommendation	View, Update, Insert, Delete	View, Update	View

Policy Manager Workflow

The following workflow describes the basic tasks that you will perform in Policy Manager for each policy that you want to create:

1. **Create the policy.** The process of creating a policy includes the following steps:
 - a. Create a [Policy record](#), which stores basic information about the policy, such as the name description.
 - b. Create a [policy diagram](#), which includes nodes and connections that represent the inputs, logic, and resulting actions for each policy.
2. **Create policy instances.** [Policy instances](#) define specific records to which the policy logic will be applied.
3. **Run the validation process.** The [validation results](#) confirm or deny that all logic is working correctly.
4. **Define an execution setting for the policy.** Determine whether you want the policy to be [executed automatically when changes are made in its associated records](#) or [according to a predefined schedule](#).
5. **Activate the policy and policy instances.** Only [active policies](#) and [active policy instances](#) will be executed.

Using the Site Map in Policy Manager

On the following pages in the Policy Manager module, a Site Map appears that looks slightly different than the Site Map that appears on other pages in the Meridium APM Framework, but the general concept is the same:

- [Policy page](#)
- [Policies Search page](#)

The following image shows an example of the Site Map that is displayed on the **Policies Search** page.



Meridium > Policy Manager > Appendix_D

The Site Map displays the following text, from left to right:

Meridium > Policy Manager > <Page Identifier>

...where:

- **Policy Manager** is a hyperlink that launches the [Policy Manager Start Page](#).
- **<Page Identifier>** is a label that identifies the current page.
 - When you are viewing the **Policy** page, this value matches the name of the current policy as it is stored in the Policy Name field in the Policy record. If the Policy is new and does not yet have a name, the label is *New Policy*.
 - When you are viewing the **Policies Search** page, the label is *Search Policies*.

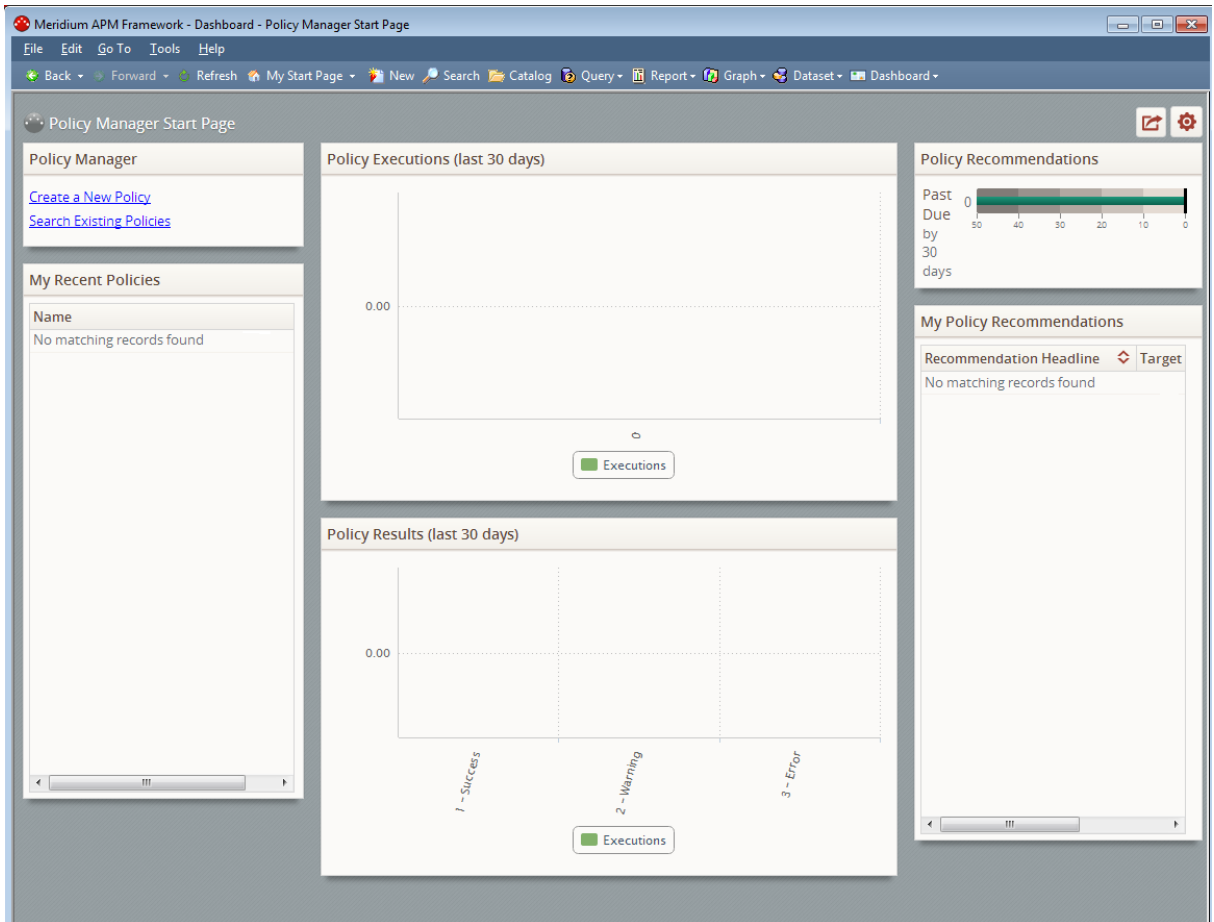
Accessing the Policy Manager Start Page

The [Policy Manager Start Page](#) provides access to the features associated with Policy Manager.

To access the Policy Manager Start Page:

- On the Meridium APM Framework menu bar, click **Go To**, point to **APM Foundation**, and then click **Policy Manager**.

The [Policy Manager Start Page](#) appears.



Aspects of the Policy Manager Start Page

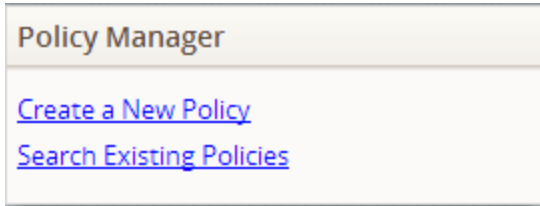
The **Dashboard -Policy Manager Start Page** displays summary information related to the Policy Manager module. The **Dashboard -Policy Manager Start Page** is different from other module start pages in that it displays the contents of the following dashboard Catalog item and its underlying queries on the **Dashboard** page: \\Public\Meridium\Modules\Policy Manager\Dashboards\Policy Manager Start Page. Throughout the documentation, we refer to the **Dashboard -Policy Manager Start Page** simply as the *Policy Manager Start Page*.

The **Policy Manager Start Page** displays the following sections by default:

- [Policy Manager](#)
- [My Recent Policies](#)
- [Policy Executions](#)
- [Policy Results](#)
- [Policy Recommendations](#)
- [My Policy Recommendations](#)

Policy Manager Section

The **Policy Manager** section is a configured **Hyperlinks** dashboard section that displays links that you can use to create new policies and search for existing policies.

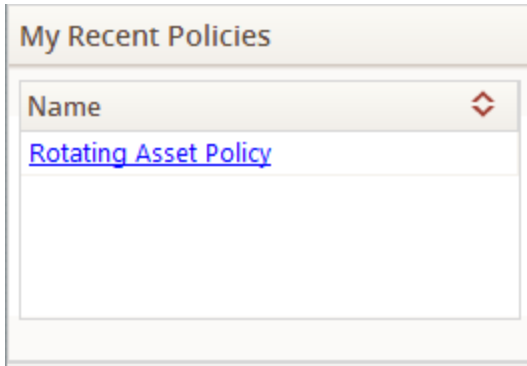


The **Policy Manager** section contains the following links:

- **Create a New Policy:** Displays the [Policy page](#), where you can [create a new policy](#).
- **Search Existing Policies:** Displays the [Policy Search page](#), where you can search for the Policy record belonging to the policy that you want to [open on the Policy page](#).

My Recent Policies Section

The **My Recent Policies** section is a configured **List - Query Result** dashboard section that displays a grid containing the name of policies that were last updated within the last 30 days by the Security User who is currently logged in (where the name is stored in the Name field in the Policy record that belongs to the policy).



The grid in this section:

- Displays one row for each policy that meets the criteria defined in this topic.
- Displays one column, **Name**, which displays the value in the Name field in the associated Policy record. Values in the **Name** column:
 - Are sorted in descending order by default, meaning that the policy that was *most recently* updated within the last 30 days by the Security User who is currently logged in is displayed in the first row in the grid.
 - Appear as hyperlinks, which you can click to [access the Policy page for the associated policy](#).
- Displays up to 20 rows per page.

Policy Executions Section

The **Policy Executions (last 30 days)** section is a [section](#) configured **Chart - Series** dashboard section that displays a bar graph that plots a maximum of the twenty policies that were executed most frequently within the last 30 days. Throughout the documentation, we refer to the **Policy Executions (last 30 days)** section simply as the ***Policy Executions section***.

The axes on the graph are configured in the following ways:

- **X-axis:** Displays one bar for each policy that has been executed within the last 30 days. The name of the bar is the value in the Name field in the associated Policy record.
- **Y-axis:** Displays a value that identifies the number of times that each policy that is plotted on the x-axis has been executed.

Below the x-axis is a legend that indicates that the graph measures the number of policy executions. You can click bars in the graph to view a details dialog box that displays the name and number of executions for a specific policy plotted on the graph.

Policy Results Section

The **Policy Results (last 30 days)** section is a configured **Chart - Series** dashboard section that displays a graph that plots the results of policies that have been [executed](#) within the last 30 days. Throughout the documentation, we refer to the **Policy Results (last 30 days)** section simply as the ***Policy Results section***.

The axes on the graph are configured in the following way:

- **X-axis:** Displays the following bars:
 - **1 - Success:** Plots the number of policies whose executions were [successful](#).
 - **2 - Warning:** Plots the number of policies whose executions resulted in a warning.
 - **3 - Error:** Plots the number of policies whose executions resulted in an error.
- **Y-axis:** Displays a value that corresponds with the number of policy executions.

Below the x-axis is a legend that indicates that the graph measures the number of policy executions. You can click bars in the graph to view a detail dialog box that displays the number of executions associated with a specific execution result.

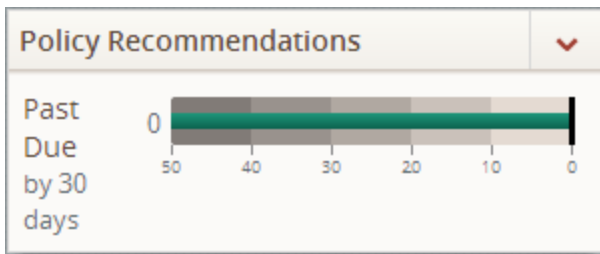
Policy Recommendations Section

The **Policy Recommendations** section is a configured **Measures - General** dashboard section that displays a measure of the number of Policy Recommendation records:

- Whose Target Completion Date field contains a value that corresponds with a date that is at least 30 days prior to today's date.
- and-
- That are *not* in the Closed state.

By default, the target value defined for this measure is 0 (zero), meaning, that *no* Policy Recommendation records should meet this criteria.

The following image shows an example of the **Policy Recommendations** section when no Policy Recommendation records meet this criteria.



My Policy Recommendations Section

The **My Policy Recommendations** section is a configured **List - Query Result** dashboard section that contains a grid that displays information about all Policy Recommendation records that are assigned to the Security User who is currently logged in.

My Policy Recommendations	
Recommendation Headline	Target Completion Date
Fixed Asset Recommendation	2012-06-22 00:00

The grid in this section:

- Displays one row for each Policy Recommendation record that is assigned to the Security User who is currently logged in.
- Displays the following columns of information that correspond with Policy Recommendation fields:

- Recommendation Headline
- Target Completion Date

Values in the:

- **Target Completion Date** column are sorted in ascending order by default, meaning that the Policy Recommendation record with the Target Completion Date value that is closest to the current day is displayed in the first row in the grid.
- **Recommendation Headline** column appear as hyperlinks, which you can click to access the [Policy page](#) for the associated policy.
- Displays up to 20 rows per page.

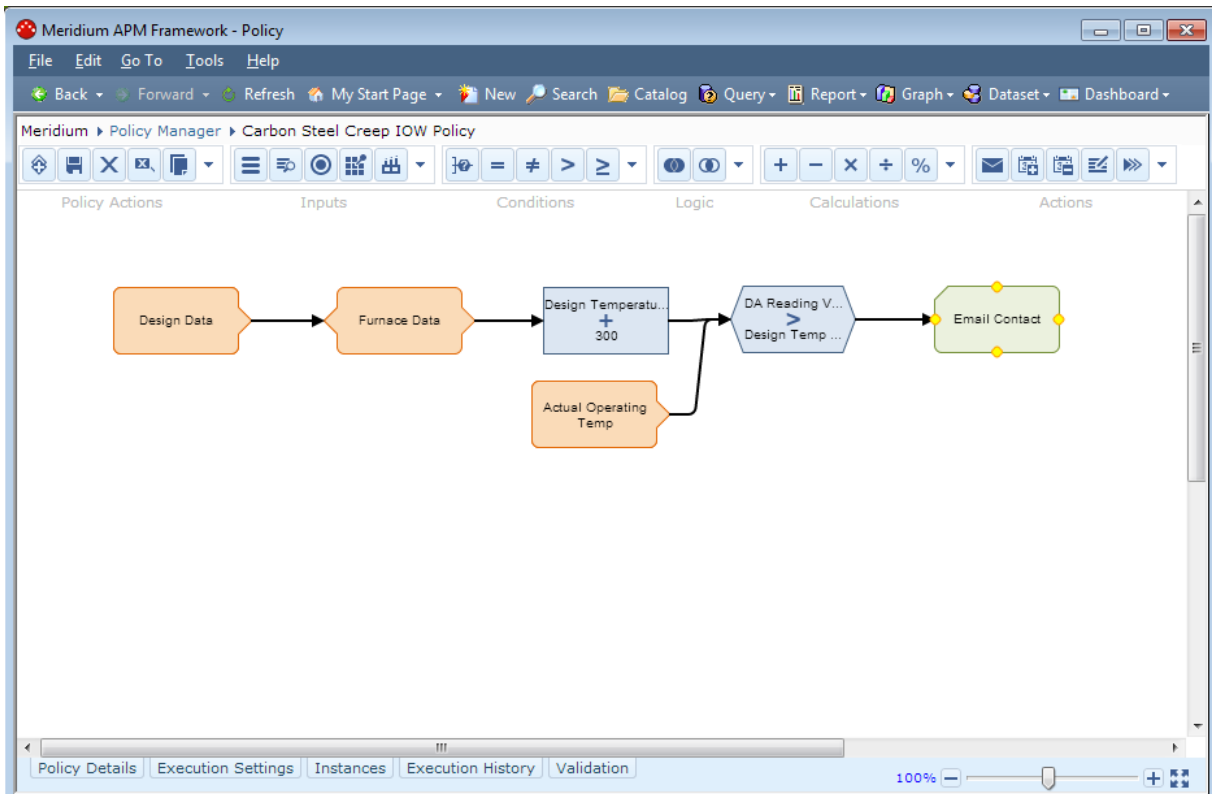
Accessing the Policy Page

Note: You must have at least View permissions on the Policy family in order to access the Policy page. If you do not, when you try to access the Policy page, an error message will appear.

To access the Policy page:

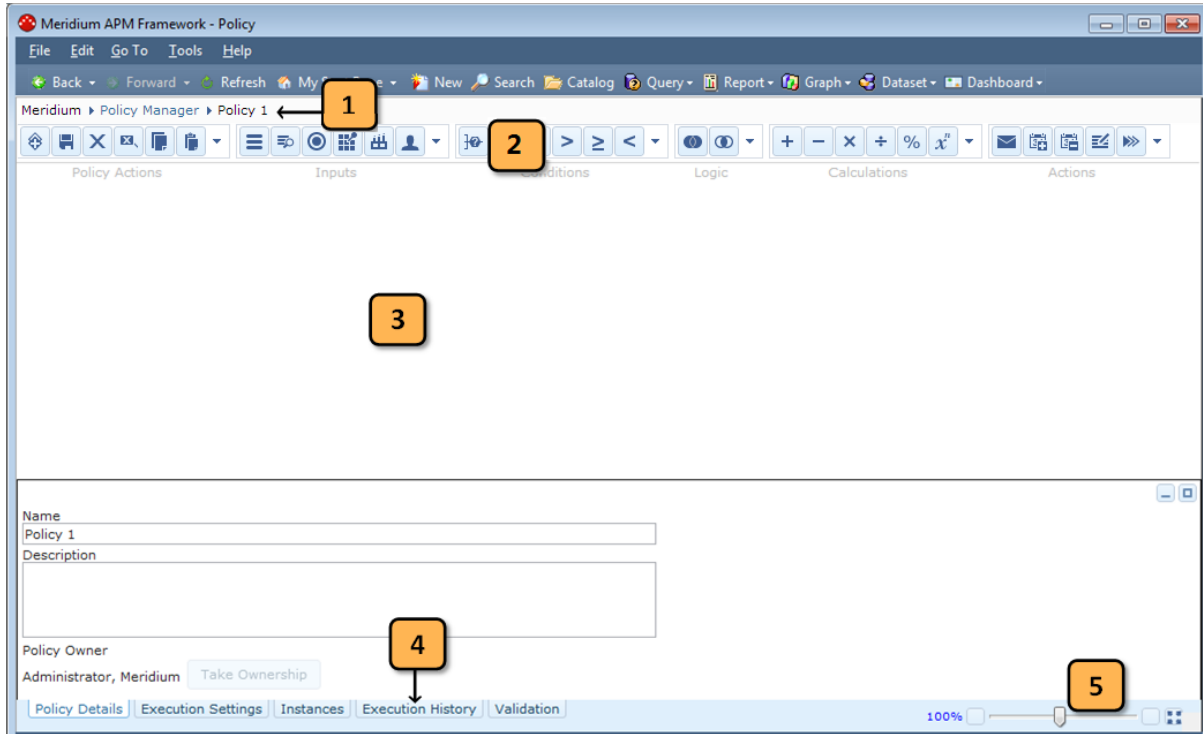
1. [Access the Policy Manager Start Page.](#)
2. In the **Policy Manager** section:
 - Click the **Create a New Policy** link to [create a new policy.](#)
 - OR-
 - Click the **Search Existing Policies** link, and [open an existing policy.](#)

The [Policy page](#) appears.



Aspects of the Policy Page



The **Policy** page appears when you [create](#) a new policy or [open an existing policy](#). Callouts have been added to the following image to highlight items on the **Policy** page.

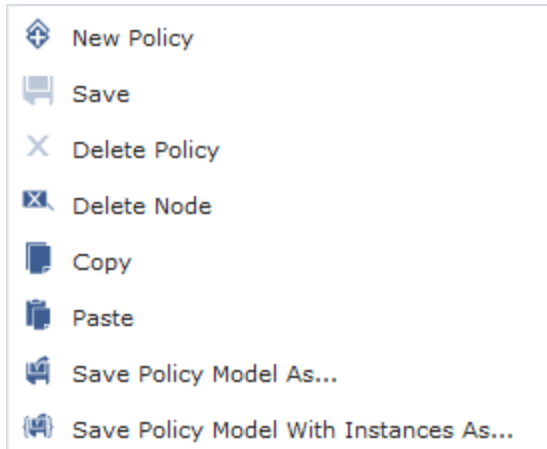


The **Policy** page contains the following items:

- **Site Map (#1 in the image):** Provides navigation to the **Policy Manager Start Page** and displays the name of the current policy.
- **Toolbars (#2 in the image):** The following toolbars are displayed, from left to right:
 - **Policy Actions:** Contains buttons that represent basic actions that you can perform when working with policies.
 - **Inputs:** Contains buttons that you can use to create Input nodes.
 - **Conditions:** Contains buttons that you can use to create Condition nodes.
 - **Logic:** Contains buttons that you can use to create *And* or *Or* nodes.
 - **Calculations:** Contains buttons that you can use to create Calculation nodes.
 - **Actions:** Contains buttons that you can use to create Action nodes, which cause certain actions to be performed when the policy is executed.

Note: You must be a member of the MI Policy Designer Security Group to create or modify policies. Otherwise, the toolbars will be disabled.



On the right side of each toolbar, the  button is displayed, which displays a list of options that correspond with the buttons on the toolbar. The options in the list are labeled. This might be useful for users whose screen size limits the number of buttons that are displayed on each toolbar or occasional users who are not familiar with the toolbar buttons. For example, consider the list that appears when you click the  button on the **Policy Actions** toolbar.



Throughout the documentation, we refer to toolbar buttons using the picture of the button as it appears on the toolbar.

- **Diagram Canvas (#3 in the image):** The main workspace in which you build the policy [diagram](#). You can [add nodes to the diagram canvas](#) using the toolbar buttons, and you can pan to other areas of the canvas by clicking and dragging the surface of the canvas or by using the scroll bars.
- **Tabs (#4 in the image):** The following tabs are displayed below the diagram canvas, from left to right:
 - **Policy Details:** Displays basic information about the policy.
 - **Execution Settings:** Provides options that you can use to [activate the policy](#), [set the execution mode for the policy](#), and [define a schedule by which to execute the policy](#).
 - **Instances:** Provides options that you can use to [create policy instances](#) and [activate](#) or [deactivate policy instances](#).
 - **Execution History:** Displays the results of policy executions.
 - **Validation:** Provides options that you can use to [validate the policy logic](#).

The following buttons are displayed at the top right of each tab.

- : Minimizes the current tab until you select it again.
- : Maximizes the current tab.
- **Zoom tool (#5 in the image):** Allows you to [zoom](#) in or out on the diagram canvas.

Policy Actions Toolbar

The **Policy Actions** toolbar contains buttons that let you perform basic actions against a policy.



The **Policy Actions** toolbar contains the contains the buttons that are described in the following table.










Image	Name	Purpose
	New Policy	After prompting you to save the current policy, opens a new policy on the Policy page.
	Save	Saves the current policy.
	Delete Policy	After asking for confirmation, deletes the current policy.
	Delete Node	After asking for confirmation, deletes the selected node and all links between it and other nodes. This button is enabled only when a node is selected.
	Copy	Creates a copy of the selected nodes.
	Paste	Pastes on the diagram canvas nodes that have been copied. This button is enabled only if you have copied a node.
	Save Policy Model As	Creates a new policy as a copy of the current policy. The new policy contains a new Policy record with a different name. Note that any instances are not copied when you use this option.
	Save Policy Model With Instances As	Creates a new policy and policy instances as copies of the current policy and its associated policy instances. Note that: <ul style="list-style-type: none"> • The new policy contains a new Policy record with a different name. • The new policy instances contain new records with the <i>same</i> name as the copied records.

Inputs Toolbar

The **Inputs** toolbar contains buttons that you can use to add [Input nodes](#) to the policy diagram.



The **Inputs** toolbar contains the contains the buttons that are described in the following table.








Image	Name	Purpose
	Entity	Creates a node that represents a Meridium APM family . You will need to identify which specific entity family the node represents using the Properties window that appears when you select the Entity node.
	Query	Creates a node that represents a query .
	Point Value	Creates a node that represents a point value .
	Measurement Location	Creates a node that represents the Measurement Location family .
	AMS Asset	Creates a node that represents the AMS Asset family .
	User	Creates a node that represents the Security User family and Human Resource family .
	Constant	Creates a node that represents a constant value .
	Health Indicator	Creates a node that represents the Health Indicator family .
	Xi Tag	Creates a node that represents the Xi Tag family .

Conditions Toolbar

The **Conditions** toolbar contains buttons that you can use to define criteria for inputs.



The **Conditions** toolbar contains the contains the buttons that are described in the following table.



Image	Name	Purpose
	Equal	Creates an Equal node.
	Not Equal	Creates a Not Equal node.
	Greater Than	Creates a Greater Than node.
	Greater Than or Equal	Creates a Greater Than or Equal node.
	Less Than	Creates a Less Than node.
	Less Than or Equal	Creates a Less Than or Equal node.
	Case	Creates a Case node .

Logic Toolbar

The **Logic** toolbar contains buttons that you can use to add [logic criteria](#).

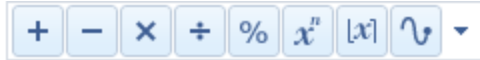


The **Logic** toolbar contains the buttons that are described in the following table.












Image	Name	Purpose
	Or	Creates an Or node.
	And	Creates an And node.

Calculations Toolbar

The **Calculations** toolbar contains buttons that you can use to [perform mathematical functions or other operations on values in your policy](#).



The **Calculations** toolbar contains the buttons that are described in the following table.









Image	Name	Purpose
	Add	Creates an Add node .
	Subtract	Creates a Subtract node .
	Multiply	Creates a Multiply node .
	Divide	Creates a Divide node .
	Remainder	Creates a Remainder node .
	Exponent	Creates an Exponent node .
	Round	Creates a Round node .
	Collection Filter	Creates a Collection Filter node .
	Last	Creates a Last node .
	Min	Creates a Min node .
	Max	Creates a Max node .
	Average	Creates an Average node .
	Sum	Creates a Sum node .
	Count	Creates a Count node .
	Threshold Statistics	Creates a Threshold Statistics node .

Actions Toolbar

The **Actions** toolbar contains buttons that you can use to [identify actions that should result from a policy execution](#).



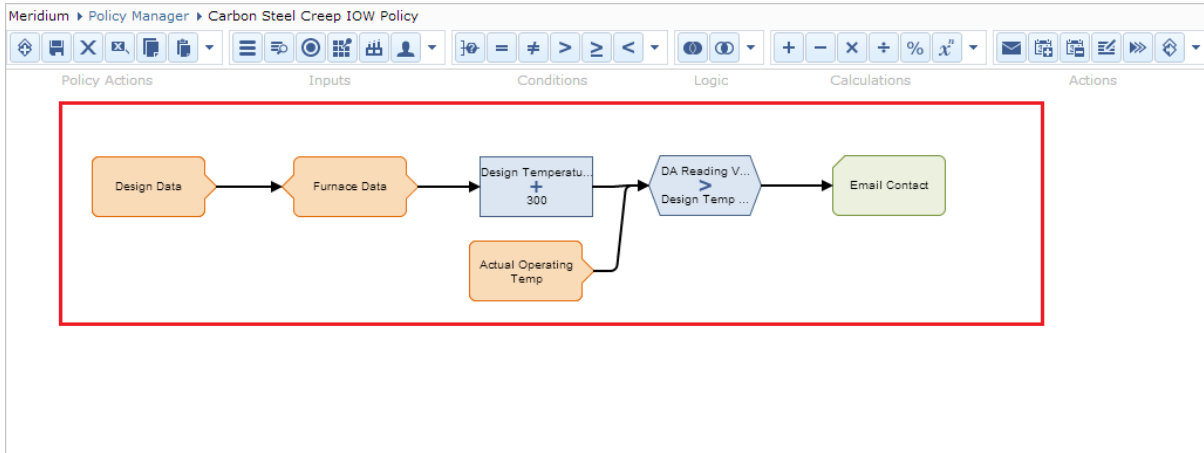
The **Actions** toolbar contains the buttons that are described in the following table.

Image	Name	Purpose
	Email Contact	Creates a node that sends an email message .
	Create Event	Creates a node that creates a Policy Event record .
	Close Event	Creates a node that populates fields in an existing Policy Event record to signify that the associated event has been closed .
	Create Recommendation	Creates a node that creates a Policy Recommendation record .
	Rule	Creates a node that executes a rule .
	Return Value	Creates a node that returns values from preceding nodes.
	Disable This Instance	Creates a node that disables the current policy instance .
	Add Value to Health Indicator	Creates an Add Value to Health Indicator node .

About the Policy Diagram

The *policy diagram* is a collection of nodes and connections that appears on the diagram canvas on the [Policy page](#). When you create a new policy from scratch, it will not have a policy diagram. The [policy diagram is built manually](#), rather, as you add nodes and connections to the diagram canvas.

In the following image, the diagram is outlined in red.



About Tabs Displayed on the Policy Page

The following tabs are displayed below the [policy diagram](#) on the [Policy page](#):

- [Policy Details](#)
- [Execution Settings](#)
- [Instances](#)
- [Execution History](#)
- [Validation](#)

Each tab displays information related to the policy that you are working with on the **Policy** page and the policy instances and executions associated with that policy. Information is displayed the tabs in various formats, with the exception of some tabs, which display information in the following common areas:

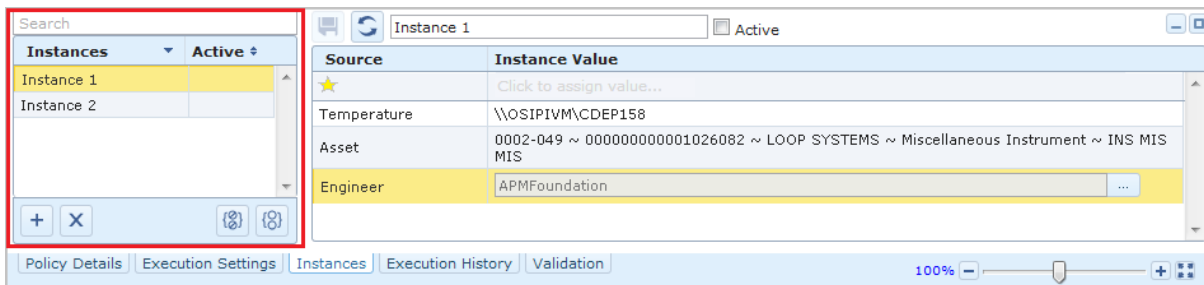
- [Search](#)
- [Details](#)

The Search Area

The *Search area* is an unlabeled area that is displayed on the left side of the following tabs:

- [Instances](#)
- [Execution History](#)
- [Validation](#)

In the following image that shows an example of the Instances tab, the Search area is outlined in red.



The Search area contains a grid that displays one row for each policy instance that is associated with the policy that you are currently viewing. When you select a row in this grid, information for the associated policy instance is displayed in the [Details area](#) on the tab. The grid contains the following columns:

- **Instances:** Displays the name of the policy instance (stored in the ID field in the Policy Instance record).
- **Active:** Displays a check mark if the policy instance is [active](#). If the policy instance is inactive, this column is empty.

Note: Additional rows are displayed in the grid on the [Execution History](#) and [Validation](#) tabs, and buttons are displayed below the grid on the [Instances](#) tab.

Above the grid, a text box is provided so that you can filter the results that are displayed in the grid according to the value that you type in the text box.

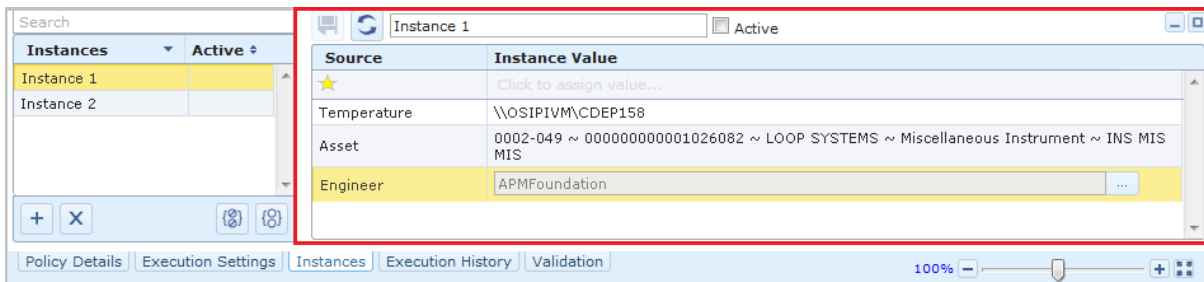
Throughout the documentation, the term Search area is used to refer to unlabeled area on the left side of the Instance, Execution History, and Validation tabs.

The Details Area

The *Details area* is an unlabeled area that is displayed to the right of the [Search area](#) on the following tabs:

- [Instances](#)
- [Execution History](#)
- [Validation](#)

In the following image that shows an example of the **Instances** tab, the Details area is outlined in red.



The Details area displays information related to the policy instance that is selected in the grid in the [Search area](#). The information and items that are displayed in the Details area varies, depending upon the tab that is selected. Throughout the documentation, the term *Details area* is used to refer to this unlabeled area on the right side of the Instance, Execution History, and Validation tabs.

Policy Details Tab

The **Policy Details** tab displays summary information about the policy that you are currently viewing.

The screenshot shows a web interface for the 'Policy Details' tab. It features a form with the following elements:

- Name:** A text box containing the value 'Fixed Asset'.
- Description:** A text box containing the value 'This policy applies to all fixed assets at the Mecklenburg plant.'
- Policy Owner:** A label 'Administrator, Meridium' followed by a 'Take Ownership' button.
- Navigation:** A horizontal bar with five tabs: 'Policy Details' (selected), 'Execution Settings', 'Instances', 'Execution History', and 'Validation'.

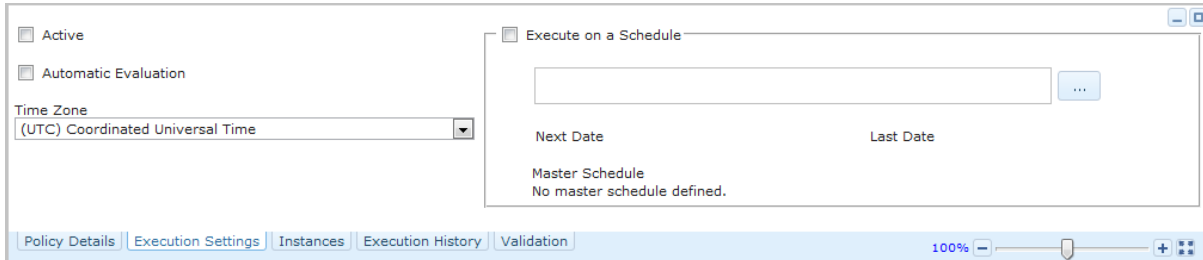
When you create a new policy, the **Policy Details** tab is displayed automatically.

The **Policy Details** tab contains the following items:

- **Name text box:** The value that you type in this text box is stored in the Name field in the associated Policy record. This text box must contain a value. If you do not type a value in this field, or if you type a value that is stored in the Name field in an existing Policy record, you will not be able to save the policy.
- **Description text box:** The value that you type in this text box is stored in the Description field in the associated Policy record. You are not required to type a value in this text box.
- **Policy Owner:** The first and last name of the [policy owner](#) is displayed below this label. When you create a new policy, you will be assigned automatically as the policy owner.
- **Take Ownership button:** Allows you to [take ownership](#) of the current policy. If you are viewing a policy for which you are not the owner, you can click this button to [take ownership](#) of the policy. The **Take Ownership** button is enabled only when you are:
 - Not the current policy owner.
 - A Super User or a member of the MI Policy Designer Security Group.

Execution Settings Tab

The **Execution Settings** tab on the [Policy page](#) contains options that you can use to specify when a policy should be executed.



The **Execution Settings** tab contains the following items:



- **Active check box:** Indicates whether or not the policy is [active](#). When the **Active** check box is:
 - *Selected*, the policy is active.
 - *Cleared*, the policy is inactive.
- **Automatic Evaluation check box:** Indicates whether or not the policy will be [executed automatically when records belonging to the policy instance are modified](#). When the **Automatic Evaluation** check box is:
 - *Selected*, the policy will be executed automatically when records are modified.
 - *Cleared*, the policy will not be executed automatically when records are modified.

In order for the policy to be executed, this check box *must* be selected if the **Execute on a Schedule** check box is *cleared*.

- **Time Zone list:** Displays a hard-coded list of time zones. The value *(UTC) Coordinated Universal Time* is selected by default. Your selection in this list is stored in the Time Zone field in the associated Policy record. The selected time zone will be used to display dates and times on the **Execution History** and **Validation** tabs. In addition, it will be used to determine the range of values that are evaluated when you specify a start end time for the execution.
- **Execute on a Schedule workspace:** Displays the following items that are used to define an execution schedule:
 - **Check box:** Displayed to the left of the **Execute on a Schedule** workspace label and indicates whether or not the policy will be [executed according to a predefined schedule](#). When the **Execute on a Schedule** check box is:
 - *Selected*, the policy will be executed according to a predefined schedule.
 - *Cleared*, the policy will not be executed according to a predefined schedule.

In order for the policy to be executed, this check box *must* be selected if the **Automatic Evaluation** check box is *cleared*.

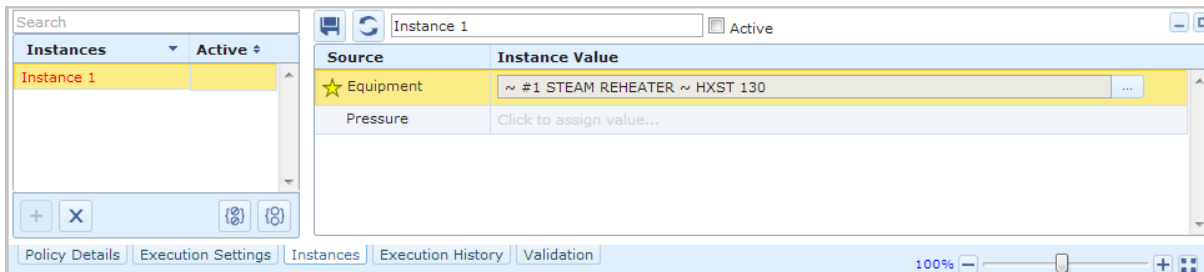
- **Box:** Displayed at the top of the **Execute on a Schedule** workspace. This box displays a summary of the schedule on which the policy will be executed automatically ([specified via the Schedule window](#)). If a schedule has not been defined, this box is empty.

To the right of this box, the  button is displayed, which opens the **Schedule** window. When a schedule summary is displayed in the box, the  button is also displayed to the right of the box. This button clears the execution schedule that is displayed in the box and any corresponding Next Date or Last Date values.

- **Next Date:** If an execution schedule has:
 - Been defined for the policy, the value that represents the next date on which the policy will be executed is displayed below this label.
 - Not been defined for the policy, no value is displayed below this label.
- **Last Date:** If an execution schedule has:
 - Been defined for the policy, the value that represents the last date on which the policy was executed is displayed below this label.
 - Not been defined for the policy, no value is displayed below this list.
- **Master Schedule:** Displays a summary of the schedule that is defined by the [scheduled item](#) in the Meridium APM Schedule Manager. If a schedule is not defined by the scheduled item in the Meridium APM Schedule Manager, the text *No master schedule defined* is displayed below this label. The execution schedule (summarized in the box) should be *less* frequent than the schedule defined in the scheduled item in order for the policy to be executed at the correct time.

Instances Tab





You can use the **Instances** tab on the [Policy page](#) to create and manage [policy instances](#).



The **Instances** tab contains the following items:

- **Search area:** Contains the grid and text box that are displayed in the standard [Search area](#). The text in the **Instances** column in the grid in this area will be colored red if the grid in the [Details area](#) contains any rows for which a record has not been specified (i.e., the row contains the text *Click to assign value*).





On the **Instances** tab, the following buttons are displayed below the grid.

	Creates a new policy instance.
	Deletes the selected policy instance.
	Deactivates all policy instances. This button is enabled when more than one policy instance is active.
	Activates all instances. This button is enabled when more than one policy instance is inactive.



- **Details area:** Contains a grid that displays one row for each node in the policy that requires values to be mapped from specific records. If a primary node has been identified in the policy diagram, a yellow star appears in the **Source** column to the left of that node name. If a primary node has *not* been identified in the policy diagram, the grid also contains one additional, empty row to represent the primary record that you can later define.

The following columns are displayed in the grid.

Column	Description
Source	<p>Displays one of the following items, depending upon what the row represents. If the row:</p> <ul style="list-style-type: none">• Represents a node in the policy diagram, the Source column displays the name of the name of the node. In addition, if the node has been identified as the primary node, a yellow star also appears in the Source column. This scenario is illustrated in the image in this topic.• Does <i>not</i> represent a node in the policy diagram, the Source column displays a yellow star, indicating that in this row, you can specify a primary record that is not associated with a node.

Column	Description
Instance Value	<p>Displays the Record ID of the selected record or the value <i>Click to assign value</i> if you have not selected a record in a row.</p> <p>When you click in this cell in:</p> <ul style="list-style-type: none"> The first row, (the row representing the primary record in the policy) the value <i>Find record</i> and the  button appear. You can use the  button to access the Search or Search <Node> window to specify the primary record in the policy. Any row <i>other than</i> the first row: <ul style="list-style-type: none"> The  button appears, which you can use to access the Search <Node> window to select a record to assign to the node. If you have specified a primary record in the policy, the text box in the cell becomes enabled, which you can use to type the Record ID of the record that you want to assign to the node. When you do this, the type-ahead functionality displays a drop-down list of <i>up to</i> five records that are linked to the primary record <i>and</i> whose Record IDs contain the value that you typed in the text box. The value <i>No suggestions</i> appears when: <ul style="list-style-type: none"> You have <i>not</i> specified a primary record in the policy. The source family does not contain at least one record that is linked to the primary record. When you have specified a primary record in the policy, but there is no relationship linking the source family and the primary record family. <p>Additionally, when a primary record has been specified that is <i>not</i> associated with a node, the  button appears in the Instance Value column in that row. This button can be used to clear the value in the Instance Column in that row.</p>

The following items are displayed above the grid in the Details area:

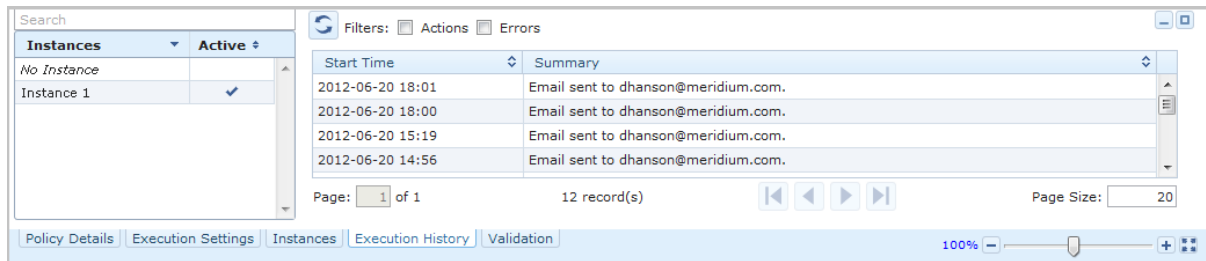
-  **button:** [Saves the policy instance](#) that is currently selected in the grid in the Search area. This button is enabled only when the selected policy instance contains unsaved changes.
-  **button:** Refreshes the grid in the Details area.
- Instance text box:** Displays the name of the policy instance that is currently

selected in the grid in the Search area (stored in the ID field in the Policy Instance record). You can modify the name of the policy instance if you want.

- **Active check box:** Indicates whether or not the policy instance that is currently selected in the grid in the Search area is active. You can select this check box to [activate the policy instance](#) or clear the check box to [deactivate the policy instance](#). You can activate a policy instance only if all nodes displayed in the grid in the Details area have records assigned to them (i.e., no rows contain the text *Click to assign value*).

Execution History Tab

The **Execution History** tab on the [Policy page](#) displays the results of policy executions.



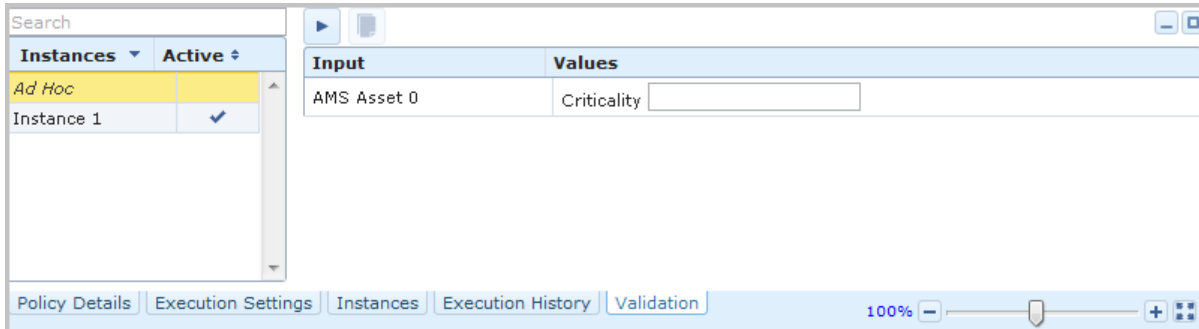
The **Execution History** tab contains the following items:

- **Search area:** Contains the grid and text box that are displayed in the standard [Search area](#). In addition to containing one row for each policy instance that is associated with the current policy, the grid contains a row with the value *No Instance* in the **Instances** column. This row exists to support policies that are executed outside of the Policy Manager module.
- **Details area:** Contains a grid that displays one row for each policy execution associated with the policy instance that is select in the grid in the Search area. When you select a row in the grid in the Details area, the policy diagram will be modified to show the details of the execution. The grid contains the following columns:
 - **Start Time:** The date and time that the execution process began. The following items are displayed above the grid:
 - **Summary:** A summary of the execution results (e.g., *Errors occurred, no action*).
 - **Refresh button:** Refreshes the grid in the Details area.
 - **Filter checkboxes:**
 - **Actions:** Displays only executions that resulted in actions.
 - **Errors:** Displays only executions that resulted in errors.
- **Page count:** Displays a text box and the text *of <#>* to the right of the text box, where *<#>* is the total number of pages available. For example, in the image in this topic, the page count indicates that page *1 of 1* is displayed.

The text box contains the value *1* by default (the first page of results is displayed by default). You can type a value in this text box to navigate to a different page of results.
- **Record count:** The text *<#> record(s)* is displayed where *<#>* is the number of policy executions found for the currently selected policy instance.
- Navigation buttons that allow you to scroll between pages.
- **Page Size text box:** The number of executions that will be displayed in the grid on a single page. This text box contains the value *20* by default.

Validation Tab

You can use the **Validation** tab to [simulate a policy execution](#) to ensure that the policy will yield the desired results.





Note: If you select the **Validation** tab for a policy that contains unsaved changes, the tab will be blank.

The **Validation** tab contains the following items:

- **Search area:** Contains the grid and text box that are displayed in the standard [Search area](#). In addition to containing one row for each policy instance that is associated with the current policy, the grid contains a row with the value *Ad Hoc* in the **Instances** column. In this grid, when you select:
 - The row containing the value *Ad Hoc* in the **Instances** column, you can modify values in the **Value** column in the grid in the [Details area](#) and [validate the policy against the values that you type manually](#).
 - A row containing a policy instance, the values displayed in the **Value** column in the grid in the [Details area](#) are disabled and cannot be modified, and you can [validate the policy against values in the records belonging to the policy instance](#).
- **Details area:** Contains a grid that displays one row for each policy node (in the current policy) that requires values to be mapped from specific records. The rows in the grid are sorted alphabetically in ascending order according to the value in the **Input** column. The grid contains the following columns:
 - **Input:** Displays the name of the node that requires a value to be mapped from specific records.
 - **Values:** Displays the name of the field from which a value will be mapped to that node and the specific value that will be mapped to that node. If a node is not being used in the policy, this column will contain the text *Not Used*.

The following buttons are displayed above the grid in the Details area:

-  **button:**[Validates the policy using the values in the grid as inputs.](#)
-  **button:**[Copies values from the record belonging to the policy instance.](#)

This button is enabled only when a policy instance is selected in the grid in the Search area, (i.e., the row containing the value *Ad Hoc* in the **Instances** column is not selected).

Types of Messages in Policy Manager


Policy Manager provides you with feedback about the operations that you are performing using a messaging system that provides minimal interruption to your workflow and identifies the location where the errors exist (e.g., which node). The Policy Manager messaging system uses the following types of messages:

- [Validation messages](#)
- [System messages](#)
- [Error messages](#)

Validation Messages

Validation messages appear in a bar at the top of the screen to provide you with information about required fields or input values. For example, when you are providing values in the **Properties** window for an Email Contact node, until an email address is provided, a validation message will appear to indicate that a valid email address format is required in the *To Address* field.

Any validation messages identifying missing *required* data appear in a red bar. Any validation messages identifying missing *optional* data appear in a yellow bar. Depending upon the message, the bar may include a **View Details** link, which displays more information about the missing information, and a **Go To Source** link, which takes you to the source of the missing information.

Validation messages continue to appear at the top of the screen until the requirements described in the message are fulfilled. If multiple validation messages are displayed at once, you can scroll through them using the  buttons. The following image displays an example of a validation message:



In this message, you can see that the policy name is missing. Because the bar is red, you can assume that the policy cannot be saved successfully until the policy name is provided.

If you were to view the **Name** text box where you can provide a policy name, you would see that it is outlined in red, which also indicates that a value is required:

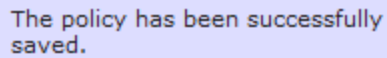
Name

Description
AMS Asset Criticality

Policy Owner
Duncan, Amy

System Messages

System messages appear in the upper right corner of the screen and then disappear after a few seconds. They are provided to confirm that an action has succeeded. For example, as shown in the image below, after you save a policy, the message *The policy has been successfully saved* appears briefly on your screen and then fades away.



The policy has been successfully saved.

Error Messages

Error messages appear in pop-up dialog boxes, similar to other error messages in the Meridium APM Framework. Usually, these messages report problems that will interrupt your workflow, such a lost connection to the database. You may also see an error message if you have not responded to all [validation messages](#) appropriately and you try to save the policy. After you click **OK** on an error message, you will need to correct the issue before you can save the policy successfully.

Refreshing the Policy Page

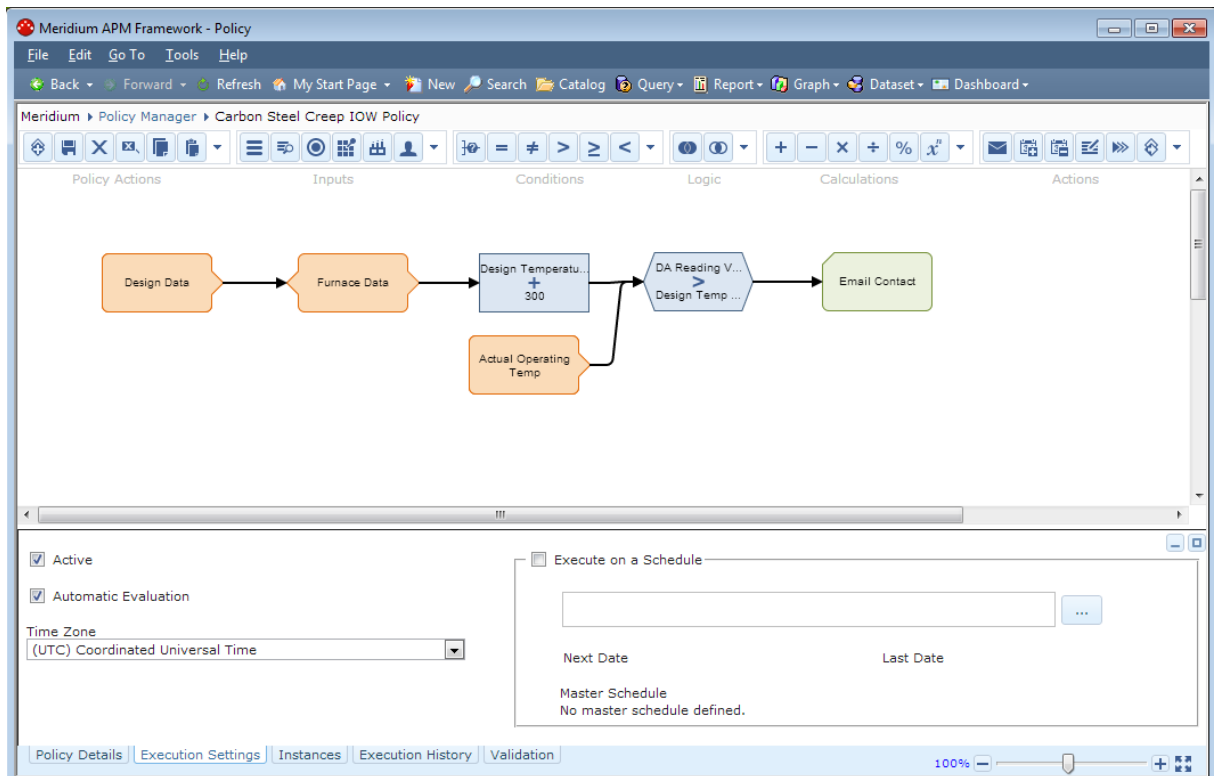
You can refresh the **Policy** page to ensure that the information that is displayed on the page for all records reflects up-to-date data from the application server. Before you refresh the **Policy** page, you should [save the policy](#) that you are working with. If you try to refresh the **Policy** page and you have unsaved modifications on the page, the **Confirm Navigation** dialog box appears, indicating that any unsaved modifications will not be retained. The following instructions assume that you are viewing the **Policy** page that does *not* contain any unsaved modifications.

To refresh the **Policy** page:

- On the Meridium APM Framework toolbar, click the **Refresh** link.

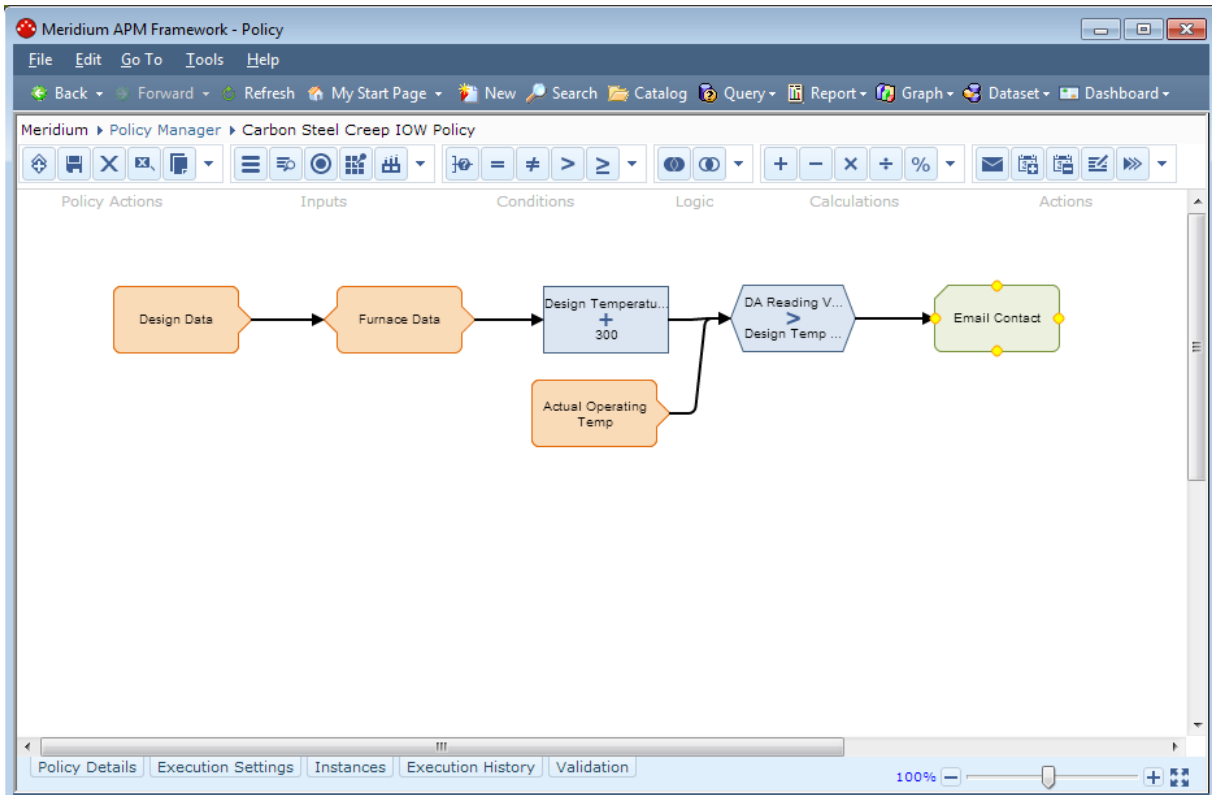
The **Policy** page is refreshed, and it is displayed according to how it appears when it is first accessed.

For example, the following image shows an example of the **Policy** page where the **Execution Settings** tab is selected.



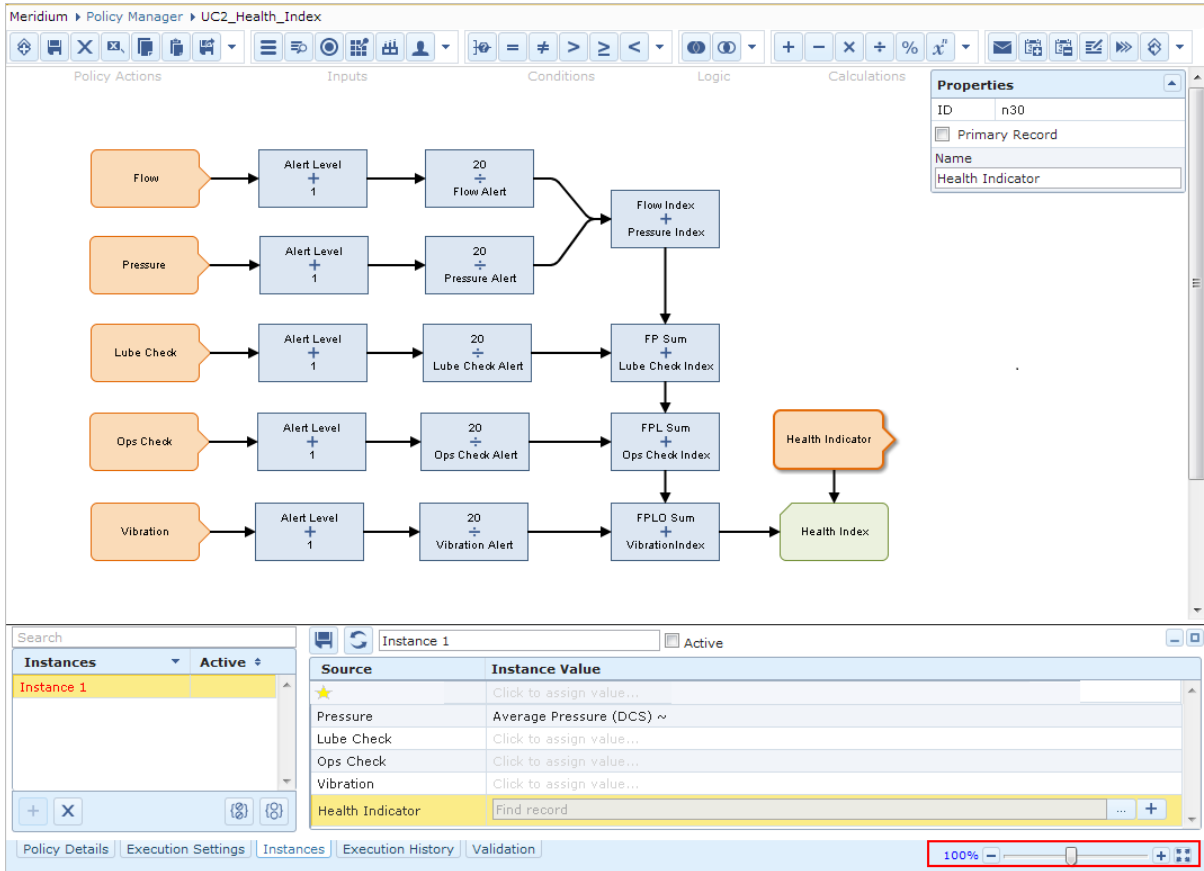
Continuing with the example, when you click the **Refresh** link, the page is displayed according to how it appears when it is first accessed, as shown in the following image.

Refreshing the Policy Page





About the Zoom Tool

As outlined in red in the following image, the zoom tool appears in the lower right corner of the Policy page.



The following table describes the features of the zoom tool.

Feature	Description
Slider bar (sliding the indicator to the right). The + button.	Zooms in on the policy diagram.
Slider bar (sliding the indicator to the left). The - button.	Zooms out on the policy diagram.

Feature	Description
The text <#>%, where <#> is a number indicating the current zoom level (e.g., )	Indicates the current zoom level. You can click the number to resets the zoom level to 100%.
The  button.	Forces the diagram canvas to display all nodes, regardless of the current size of the diagram canvas (e.g., if the size of the Meridium APM Framework application has been decreased by dragging the window borders).

Note that the following keyboard shortcuts can also be used to zoom in or out on the policy diagram:

- **Ctrl+Down Arrow:** Zooms in on the policy diagram.
- **Ctrl+Up Arrow:** Zooms out on the policy diagram.

Creating Policies from Scratch

The following instructions provide details on creating a Policy record, which is the first step in the [Policy Manager workflow](#). After you have created the Policy record, you must [create a policy diagram](#) to complete the policy.

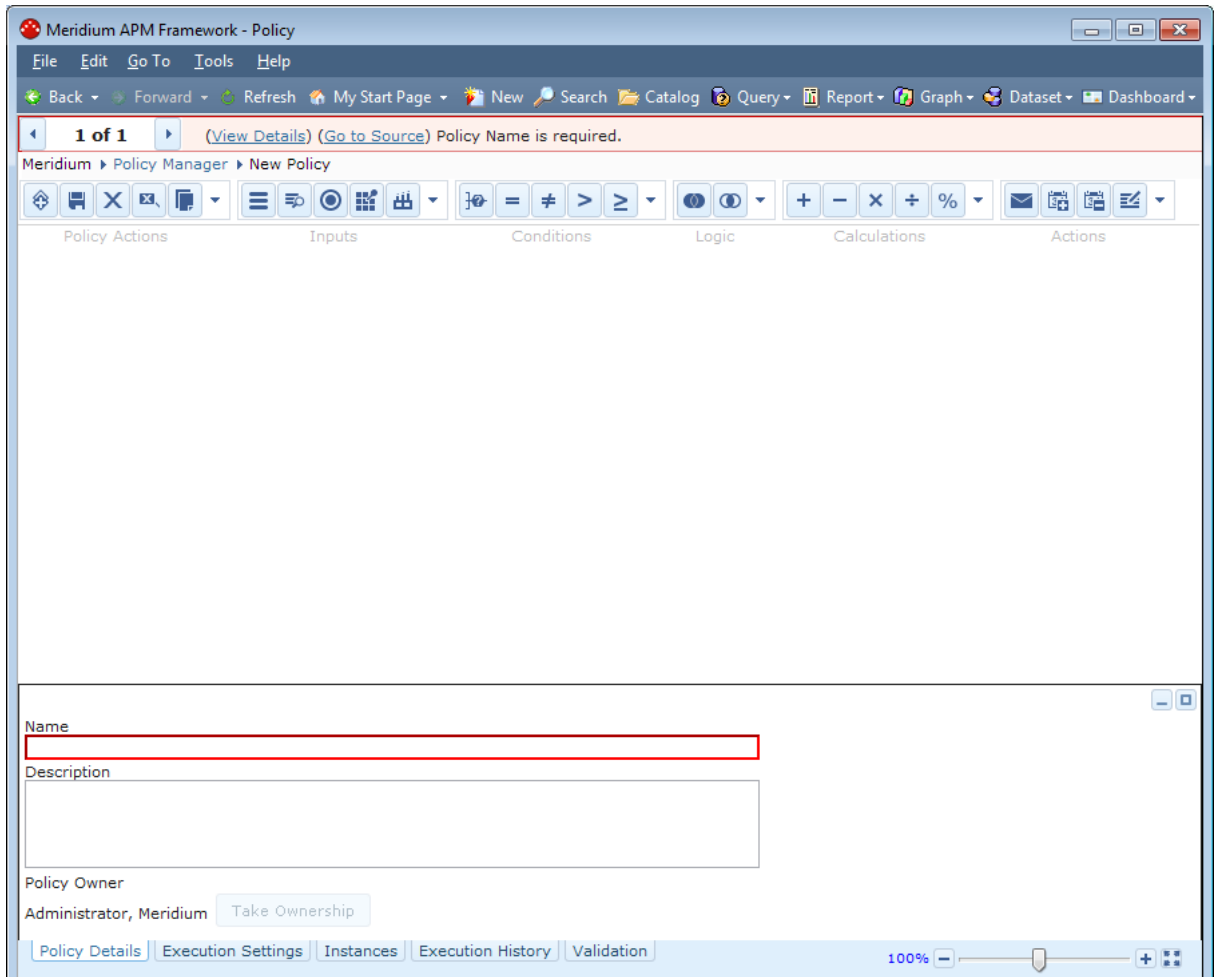
Note that you *cannot* create a new Policy record via the Record Manager because the Policy family is locked down (i.e., the **Family is Managed By Product Module** check box is *selected* in the Configuration Manager). As a result, if you attempt to create a new Policy record by clicking the **New** button on the Meridium APM Framework toolbar, you will find that the Policy family does not appear in the list of available families on the **Select Family** dialog box.

Note: You must be a member of the MI Policy Designer Security Group to create a new policy.


To create a new policy:

1. On the [Policy Manager Start Page](#), in the **Policy Manager** section, click the **Create a New Policy** link.

The **Policy** page appears.



The name of the currently logged in Security User is displayed below the **Description** text box in the format *Last Name, First Name*.


2. On the **Policy Details** tab, in the **Name** text box, type a name for the policy. This value is required and must be unique.
3. On the **Policy Actions** toolbar, click the  button.

The Policy record belonging to the policy is saved to the database.

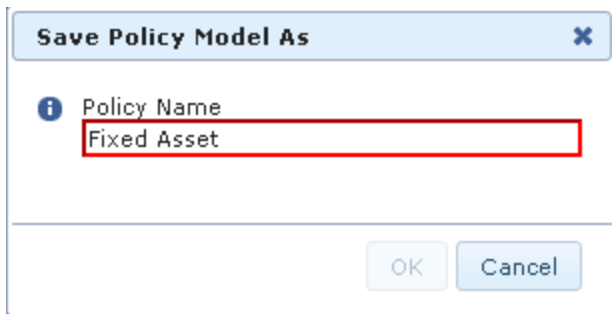
Copying a Policy

In addition to [creating a policy from scratch](#), you can also create new policies by copying existing policies. The following instructions explain how to copy a policy.

To copy a policy:

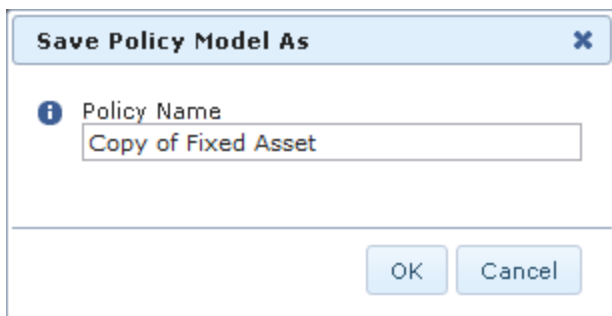
1. Access the [Policy page](#) for the policy you want to copy.
2. On the **Policy Actions** toolbar, click the  button.

The **Save Policy Model As** dialog box appears, displaying the name of the current policy in the **Policy Name** text box.



3. In the **Policy Name** text box, type a name for the new policy. The value must be unique.

The **OK** button becomes enabled.




4. Click **OK**.

The **Policy** page appears, displaying the new policy.

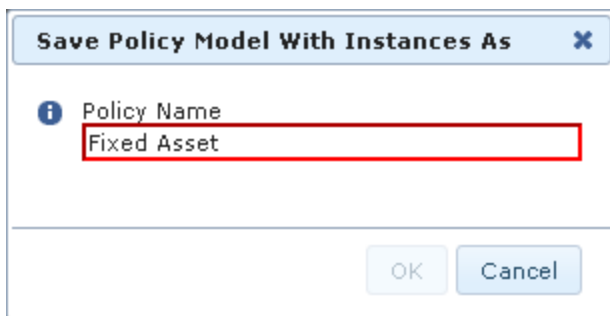
Copying a Policy and its Associated Instances

In some cases, you may want to create a new policy and policy instances by copying an existing policy and its associated instances.

To copy a policy and its associated instances:

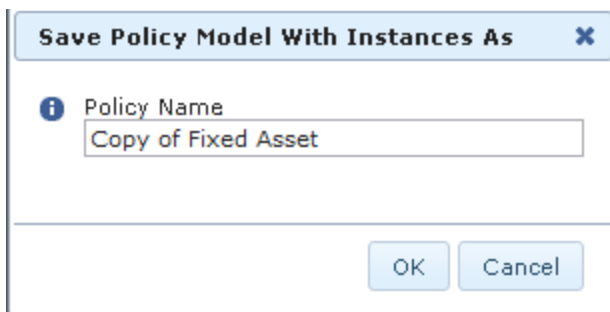
1. Access the [Policy page](#) for the policy you want to copy.
2. On the **Policy Actions** toolbar, click the  button.

The **Save Policy Model With Instances As** dialog box appears, displaying the name of the current policy in the **Policy Name** text box.



3. In the **Policy Name** text box, type a name for the new policy. The value must be unique.

The **OK** button becomes enabled.



4. Click **OK**.

The **Policy** page appears, displaying the new policy and policy instances.

Opening Existing Policies

The following instructions provide details on opening an existing policy in Policy Manager. Note that you can also open existing Policy *records* in the Record Manager, but because the Policy family is locked down (i.e., the **Family is Managed By Product Module** check box is *selected* in the Configuration Manager), you will not be able to modify any of the information for existing Policy records via Record Manager. When you open a Policy record in the Record Manager, a link will appear at the top of the datasheet area that will take you to the **Policy** page.

To open an existing policy:

1. On the [Policy Manager Start Page](#), in the **Policy Manager** section, click the **Search Existing Policies** link.

The [Policies Search page](#) appears, displaying a list of *all* existing policies.

Policy Name	Policy Description	Active
Appendix B	Appendix B	Yes
Appendix D	Appendix D	Yes
Appendix E	Appendix E	Yes
Appendix F	appendix F	Yes
Appendix G	appendix G	Yes
Appendix H	Appendix H	Yes
Appendix I	Appendix I	Yes
PRD Strategies	Relief Valve & Rupture Disk Strategies	Yes

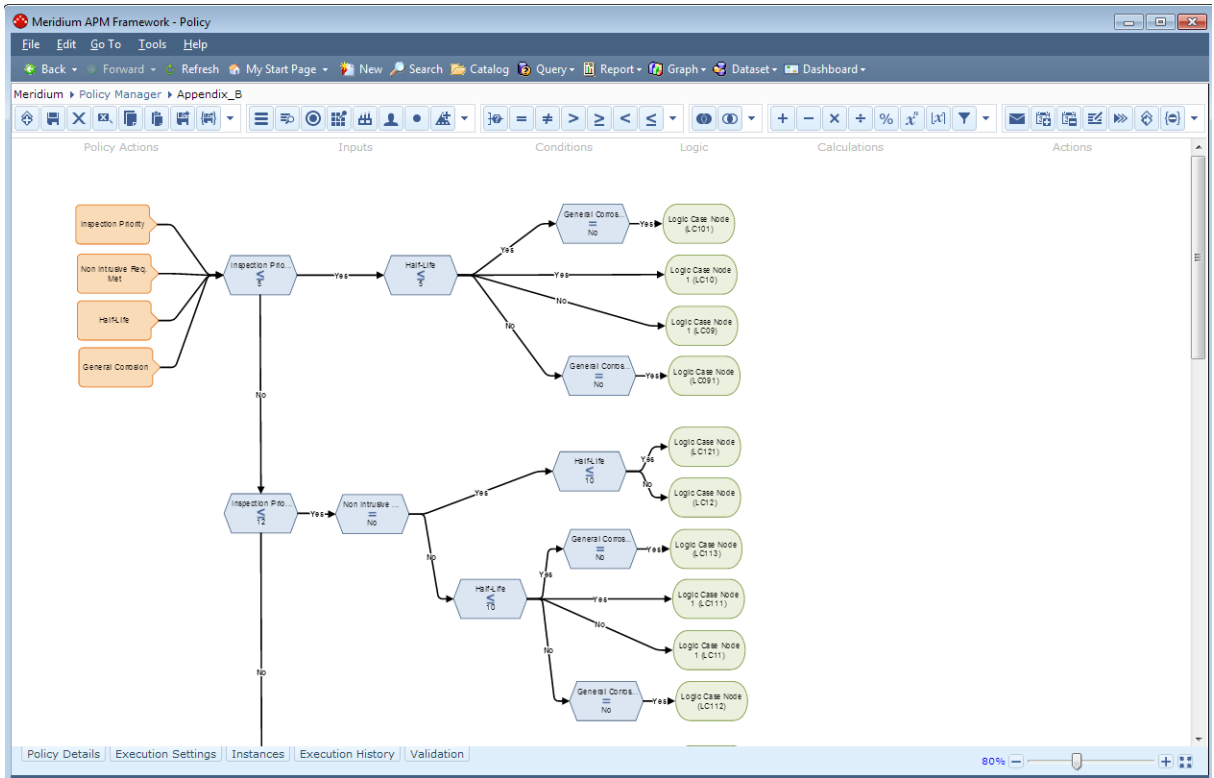
2. If desired, type a keyword in the **Filter** text box.
3. If you want the list to display only active policies, select the **Show only active policies** check box.
3. Click the **Search** button.

The grid displays policies whose Policy record contains the specified keyword in the Name or Description field.

4. In the **Policy Name** column, click the name of the policy that you want to open.

Opening Existing Policies

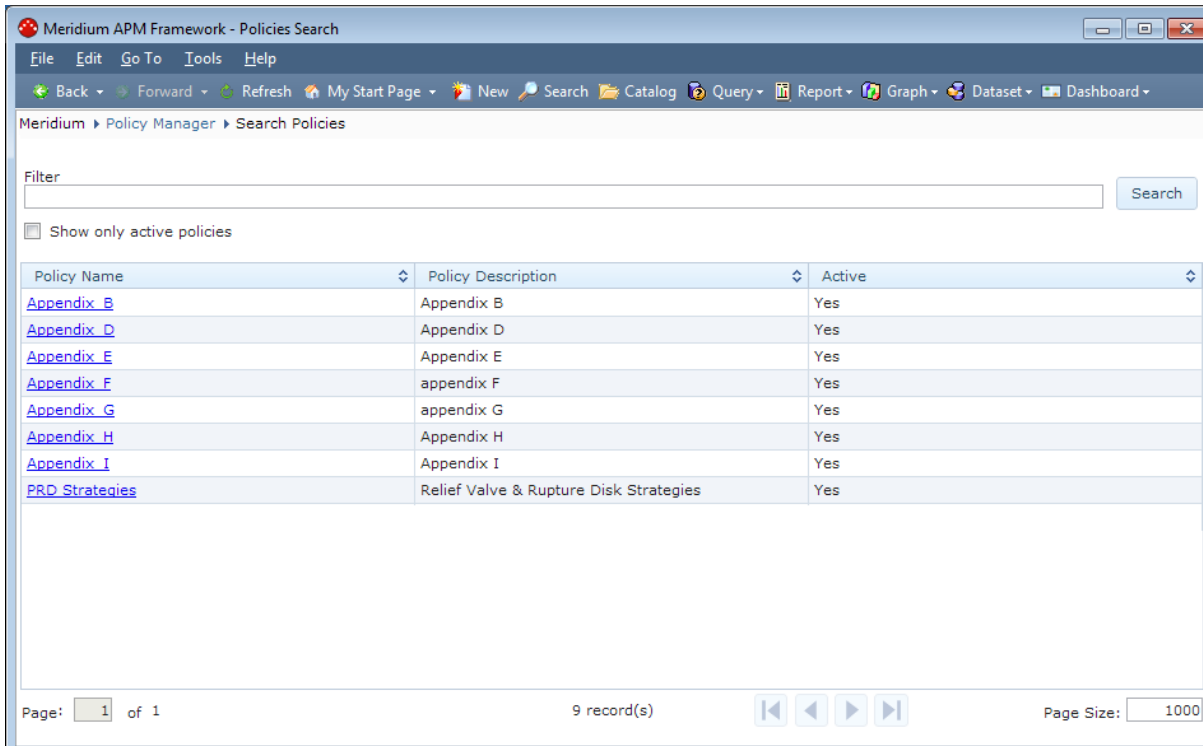
The **Policy** page appears, displaying the selected policy. In the following image, the policy *Appendix_B* is displayed on the **Policy** page.



Note: If you are a Super User or a member of the MI Policy Designer Security Group and want to modify an existing policy, you must [take ownership](#) of the policy.


Aspects of the Policies Search Page

The **Policies Search** page uses the *Policies* query, which is located in the Catalog folder \\Public\Meridium\Modules\Policy Manager\Queries, to retrieve all existing policies.



The **Policies Search** page contains the following items:

- **Site Map:** Provides navigation to the Policy Manager Start Page.
- **Filter text box:** Lets you filter the results based upon a keyword in this text box.
- **Show only active policies check box:** Lets you show only active policies in the results. If this check box is cleared, both active and inactive policies will be displayed.
- **Grid:** Displays all policies that meet the filtering criteria. The grid contains the following columns:
 - **Policy Name:** The name of the policy.
 - **Policy Description:** A description of the policy.
 - **Active:** A value that indicates whether or not the policy is active (*Yes* or *No*).

You can sort the contents of each of the columns using the  button.

- **Navigation pane:** Allow you to navigate between pages if more results exist than the grid can display in a single view. The following navigational tools are available:
 - **Page count:** Displays a text box and the text *of <#>* to the right of the text box, where *<#>* is the total number of pages available. For example, in the image in this topic, the page count indicates that page *1 of 1* is displayed.

The text box contains the value *1* by default (the first page of results is displayed by default). You can type a value in this text box to navigate to a different page of results.

- **Record count:** The text *<#> record(s)* is displayed where *<#>* is the number of policy executions found for the currently selected policy instance.
- Navigation buttons that allow you to scroll between pages.
- **Page Size text box:** The number of executions that will be displayed in the grid on a single page. This text box contains the value *20* by default.

About Policy Ownership

A *policy owner* is the Meridium APM Security User who is responsible for a given policy. Only the policy owner can save changes to a policy. Each policy is owned by a particular Security User.

In order to be a policy owner, you must be a Meridium APM Super User or a member of the MI Policy Designer Security group.

This section of the documentation provides details on:

- [Taking ownership of a policy](#)
- [Viewing the ownership history for a policy](#)

Note: Exclusive privileges are granted to policy owners. Meridium APM Super Users do not have the same privileges as policy owners.

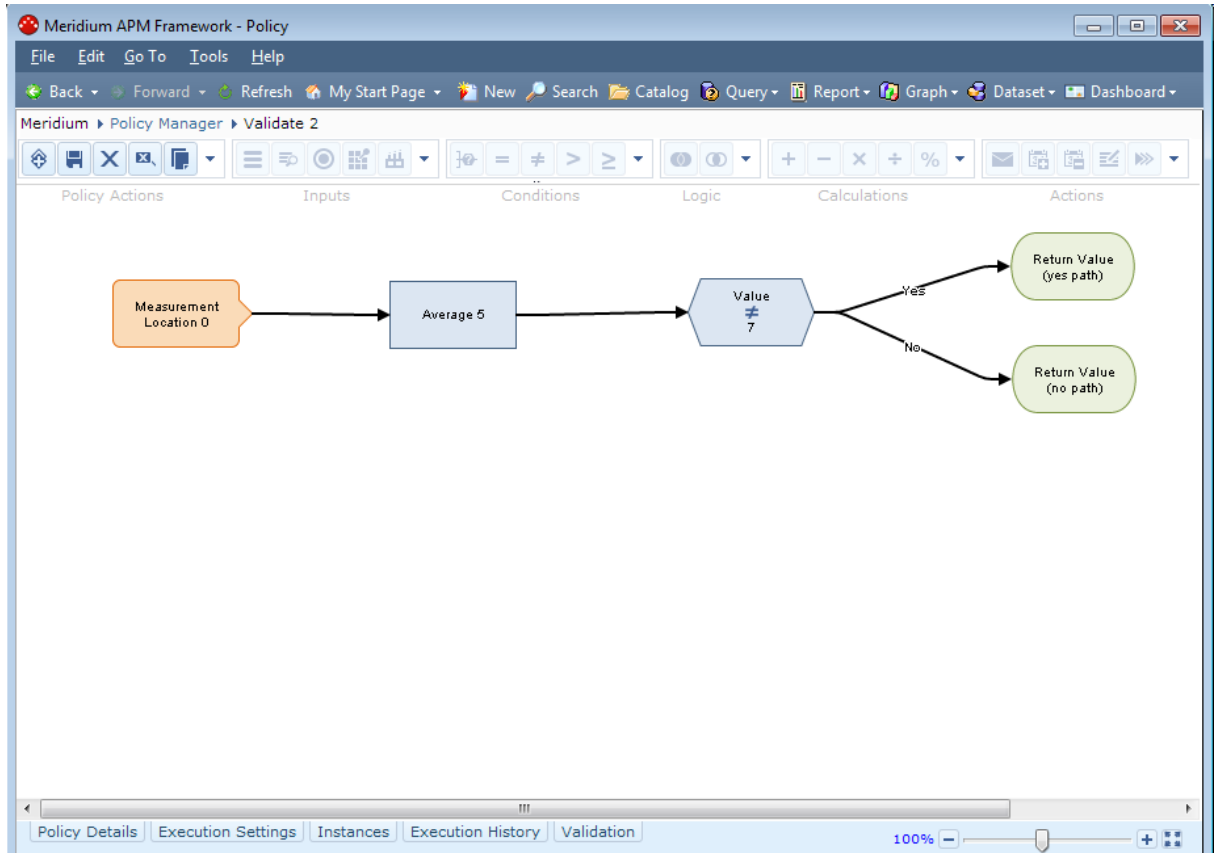
Taking Ownership of a Policy

Some options for working with policies are available only to the [policy owner](#). If you want to access an option that is unavailable to you, you can take ownership if the policy.

To take ownership of a policy:

1. [Open the policy](#) for which you want to become the owner.


The **Policy** page appears.



2. On the **Policy Details** tab, click the **Take Ownership** button, which is outlined in red in the following image.

The screenshot shows a web interface for policy management. It features three main text input fields: 'Name' containing 'UC3_Health_Index', 'Description' containing 'Use Case 3 - Asset Health Index' and a longer paragraph below it, and 'Policy Owner' containing 'DeFalco, C'. A blue button labeled 'Take Ownership' is positioned to the right of the 'Policy Owner' field and is highlighted with a red rectangular border. Below these fields is a horizontal toolbar with five tabs: 'Policy Details', 'Execution Settings', 'Instances', 'Execution History', and 'Validation'. The 'Policy Details' tab is currently selected and highlighted in light blue.

The last and first name of the policy owner that appears to the left of the **Take Ownership** button changes to reflect your last name and first name (as they are stored in your Security User record), and the policy becomes enabled.

3. On the **Policy Actions** toolbar, click the  button.

The policy is saved, and you are now the policy owner.

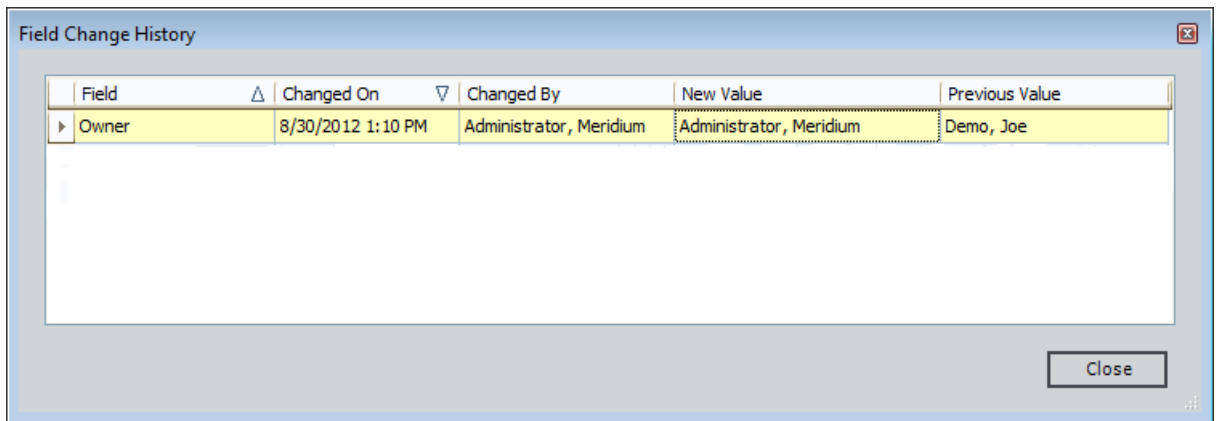
Viewing Policy Ownership History

You can view the ownership history for a policy to determine if and when other Security Users have been the owner of a particular policy.

To view the ownership history of a policy:


1. Using the Search Tool, perform a search for the Policy record associated with the policy for which you want to view ownership history, and open the record in Record Manager.
2. Access the **Field Change History** dialog box.

The **Field Change History** dialog box appears, displaying ownership history for the policy. The following image shows an example of a **Field Change History** dialog box that indicates that *Administrator, Meridium* [took ownership](#) on ownership of the given policy from *Demo, Joe* on *8/30/12 at 1:10 PM*.



Saving Changes to a Policy

To save changes to a policy:

- While [viewing a policy](#) that contains unsaved changes, on the **Policy Actions** toolbar, click the  button.

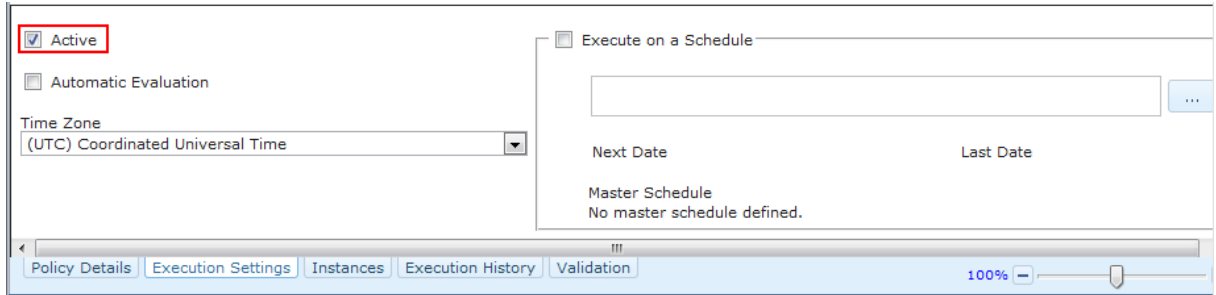
Your changes are saved to the database.

Activating Policies


The following instructions assume that you are viewing the **Policy** page for the policy that you want to activate.

To activate a policy:

1. On the **Execution Settings** tab, select the **Active** check box.



The policy is activated.

2. On the **Policy Actions** toolbar, click the  button.

The policy is saved.


Deactivating Policies

The following instructions assume that you are viewing the **Policy** page for the policy that you want to deactivate.

To deactivate a policy:

1. On the **Execution Settings** tab, clear the **Active** check box.

The policy is deactivated.

2. On the **Policy Actions** toolbar, click the  button.

The policy is saved.

Note: When you deactivate a policy, none of the policy instances that are associated with the policy will be executed.

Deleting Policies

The following instructions assume that you are viewing the **Policy** page for the policy that you want to delete. Only the [policy owner](#) can delete a policy.

To delete a policy:

1. On the **Policy Actions** toolbar, click the  button.

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

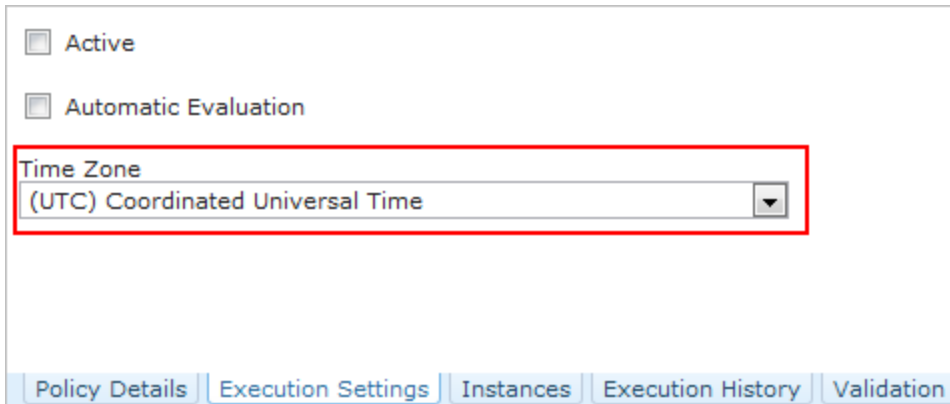
2. Click the **Delete** button.

The policy is deleted from the database.

About the Time Zone Setting

On the **Execution Settings** tab, you can select a time zone that will be used by the policy.

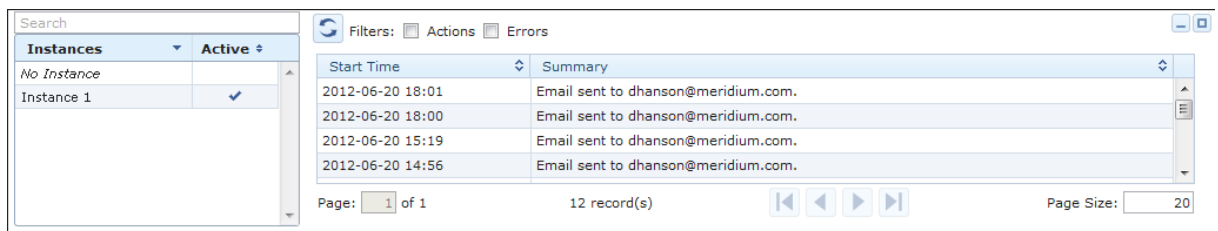
Note: The policy Time Zone setting will be used *instead of* the Windows time zone on the Meridium APM Application Server, the Windows time zone on the Meridium APM Framework workstation, and the value in the Timezone field in the Security User record of the currently logged in Security User or the Security User who is the current policy owner.



The screenshot shows the 'Execution Settings' tab in a software interface. At the top, there are two checkboxes: 'Active' and 'Automatic Evaluation', both of which are unchecked. Below these is a dropdown menu labeled 'Time Zone' which is currently set to '(UTC) Coordinated Universal Time'. The dropdown menu is highlighted with a red rectangular border. At the bottom of the tab, there are five sub-tabs: 'Policy Details', 'Execution Settings' (which is active), 'Instances', 'Execution History', and 'Validation'.

The Time Zone setting is used to determine:

- The dates and times that are displayed on the **Execution History** tab, as shown in the following image.



The screenshot shows the 'Execution History' tab. It features a search bar at the top left, a filter section with 'Actions' and 'Errors' checkboxes, and a table of execution records. The table has two columns: 'Start Time' and 'Summary'. The records show a sequence of email messages sent to dhanson@meridium.com. The start times are 2012-06-20 18:01, 2012-06-20 18:00, 2012-06-20 15:19, and 2012-06-20 14:56. Below the table, there is a pagination control showing 'Page: 1 of 1' and '12 record(s)', along with navigation arrows and a 'Page Size: 20' dropdown.

Start Time	Summary
2012-06-20 18:01	Email sent to dhanson@meridium.com.
2012-06-20 18:00	Email sent to dhanson@meridium.com.
2012-06-20 15:19	Email sent to dhanson@meridium.com.
2012-06-20 14:56	Email sent to dhanson@meridium.com.

- The dates and times that are displayed in validation messages.
- The dates and times that are displayed in email messages that are generated by policies.
- The dates and times that are displayed in Policy Recommendation records that are generated by policies.
- Which times to evaluate when you specify a [relative date range](#).

For example, suppose that you want process historian reading values to be evaluated using the following start and end times:

- **Start time:** *today + 4 hours*
- **End time:** *today + 8 hours*

The screenshot shows a 'Properties' dialog box with the following fields:

- ID: n2
- Name: Threshold Statistics 2
- Collection: Field
- Xi Tag 3 (dropdown)
- HDA Readings (dropdown)
- Start Time: Field, (highlighted with a red box)
- End Time: Field, (highlighted with a red box)

Because *today* is interpreted as 12:00:00 A.M. of the current day, the times shown in the image above are translated as:

- **Start time:** 4:00 A.M. today
- **End time:** 8:00 A.M. today

These times are specific to the time zone defined for the policy and will be the same across time zones. For example, using UTC time as the baseline, consider the following time zones:

- Eastern Time (US & Canada) = UTC - 5 hours
- Pacific Time (US & Canada) = UTC - 8 hours

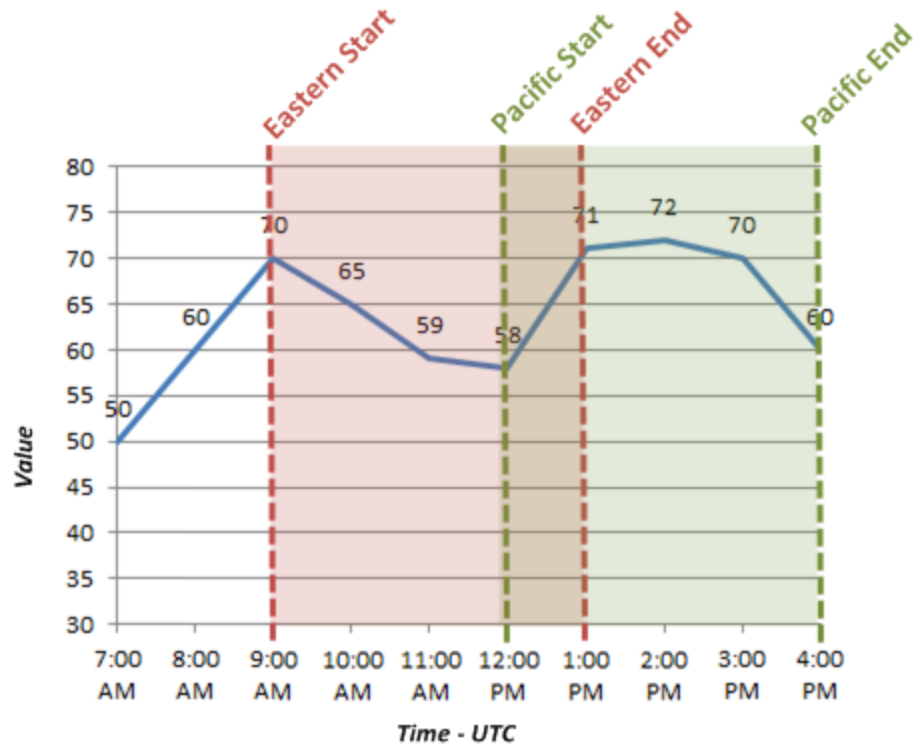
Using the relative start time of *today + 4 hours*, if the Time Zone setting specifies:

- (*UTC-5:00*) *Eastern Time (US & Canada)*, the Meridium APM system will evaluate values starting at 4:00:00 A.M. *Eastern*, which is the equivalent of 9:00:00 A.M. *UTC*.
- (*UTC-5:00*) *Pacific Time (US & Canada)*, the Meridium APM system will evaluate values starting at 4:00 A.M. *Pacific*, which is the equivalent of 12:00:00 P.M. *UTC*.

These times are represented by the following image, where:

- The (*UTC-5:00*) *Eastern Time (US & Canada)* start and end time is represented by the *red* shaded region.
- The (*UTC-5:00*) *Pacific Time (US & Canada)* start and end time is represented by the *green* shaded region.

About the Time Zone Setting



In other words, if you specify a date and time that is relative to 12:00:00 A.M. of a specific day, the date range criteria and the policy time zone work together to determine the range of data that is included in the evaluation. If two policies are set up to evaluate the exact same data but those policies use different time zones, each of those policies will end up evaluating a different set of data.

Policy Diagram Workflow

The following workflow describes the basic tasks that you need to perform to create a policy diagram:

1. **Add nodes to the diagram canvas.** Most policy diagrams consist of one or more [input](#) nodes, various [calculation](#), [condition](#), and [logic](#) nodes, and one or more [action](#) nodes.
2. **Connect the nodes** in the policy diagram.
3. **Configure the properties for each node.** In the **Properties** window for each node, you can specify a [name and label for the node](#), [define input values](#), and configure additional settings that are specific to each node.

Before you begin, it is important to understand some principles for [working with the policy diagram](#).

About Working with the Policy Diagram

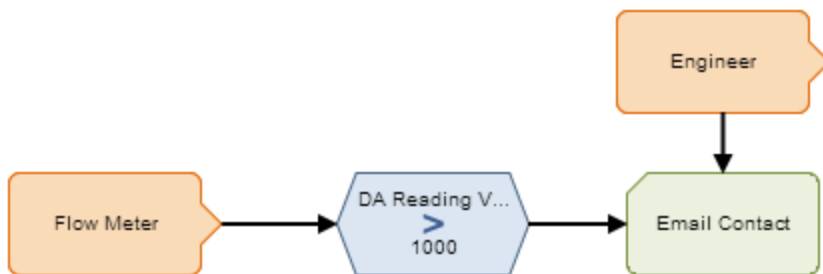
The following principles apply to working with the policy diagram:

- Policy diagrams do not reference specific *records*. Rather, they can contain nodes that reference *families*. You must use [policy instances](#) to identify the individual records whose values are evaluated when the policy is executed.
- The first node in a policy diagram must be an [Input node](#) other than a Query node.
- Every node, with the exception of Action nodes, has *outputs* that successor nodes can use as *inputs*. You must specify inputs for these nodes in the **Properties** window that appears when you select each node in the policy diagram.

Note: Outputs and inputs may represent either a single value *or* a collection of values. The types of outputs that each node generates and the types of inputs that each node accepts is different for each node. When setting up a policy diagram, you must use corresponding input and output types.

- A node can use an input from any predecessor node, even if the nodes are not directly connected.
- Any number of nodes can use an input from the same predecessor node.
- A policy diagram can often be arranged in various configurations without impacting the execution results. You should arrange the policy diagram in the configuration that provides the best visual representation of the policy.
- Connections that start at a Condition or Logic node have [properties](#) that you can use to create separate logic paths depending upon whether the condition is true or false. If you do not define the logic for a connection, a value of Yes is assumed but does not appear on the diagram.

To illustrate the principles for working with a policy diagram, consider the following nodes and connections.



In this example diagram, the node named *Flow Meter* is an Xi Tag node which represents an Xi Tag record. When this policy is executed, if the DA Reading Value that is associated with the Xi Tag record is greater than 1000, the Meridium APM system will send an email message to the email address that is specified in the Human Resource record that is associated with the *Engineer* User node.

In this example, you can see that:


- The *Flow Meter* and *Engineer* nodes represent families. The specific records whose values are evaluated are determined by policy instances.
- The *Condition* node includes:
 - The value *DA Reading Value*, which comes from the *Flow Meter* node.
 - The condition $>$.
 - The value *1000*, which is defined manually within the *Condition* node.
- The *Email Contact* node can use an input from the *Flow Meter* node even though the two nodes are not *directly* connected.
- The *Email Contact* node and the *Condition* node can both use inputs from the *Flow Meter* node.
- The execution results of the policy would be identical even if the *Engineer* node were connected to the *Flow Meter* node or the *Condition* node. However, the current configuration provides a clear visual representation of the policy because the *Email Contact* node is the only node in the diagram that uses an input value from the *Engineer* node.
- Because the logic is not defined for the connection between the *Condition* node and the *Email Contact* node, a value of *Yes* is assumed. This means that an email message is sent only if the condition defined in the *Condition* node is *true*.

[Click here](#) for additional information about this policy diagram example.

Adding Nodes to the Policy Diagram

The first step in [creating a policy diagram](#) is to add nodes to the diagram.

To add a node to the policy diagram:

1. On the [Policy page](#), on the toolbar containing the button representing the node that you want to add, click the desired button. For example, if you wanted to add an Entity node, you would click the  button.

The node is added to the diagram canvas in a default location.

2. Position the node in the desired location by selecting it, dragging it, and dropping it in the desired location. You can also move a selected node using the arrow keys.


CEHint: You can also add a node to the policy diagram by selecting the desired button on the toolbar, dragging it to the location where you want the node to appear, and dropping it. The corresponding node will appear on the diagram in that location.

Connecting Nodes in a Diagram

After you [add nodes](#) to the policy diagram, you must connect the nodes before you can fully define the policy logic.


To connect two nodes in a diagram:

1. On the predecessor node, point to the yellow circle on the side from which you want to start the connection.

The pointer changes to .



2. Click the yellow circle, and drag your mouse in the direction that you want to create the connection.

The pointer changes to , and a line is drawn automatically in the direction that you dragged the mouse.



3. Continue dragging the mouse until the line reaches the desired destination (i.e., the circle on the node that will serve as the end of the connection). The connection will be completed automatically.



4. Release your mouse button to set the connection.

The following limitations apply to connections:

- A connection cannot be created in a circular path (e.g., if Node 1 is connected to Node 2, and Node 2 is connected to Node 3, you cannot connect Node 3 to Node 1).
- A node cannot be connected to itself.
- You cannot connect one node to another node more than once.

Note: Connections that start at Condition or Logic nodes have [properties that you can use to create separate logic paths](#) depending upon whether the condition is true or false.

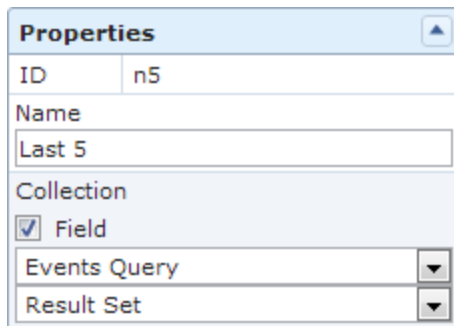
Configuring Node Properties

After you have connected nodes in the policy diagram, you must configure various properties for each node in order to complete the policy diagram. All nodes except for [Logic nodes](#) have properties that must be configured.

To configure the properties of a node:

1. In the policy diagram, select the node whose properties you want to configure.

The **Properties** window for the node appears. The following image shows an example of what the **Properties** window looks like for a Last node.



The screenshot shows a 'Properties' dialog box for a node. It has a title bar with the word 'Properties' and a close button. The dialog is divided into several sections: 'ID' with the value 'n5', 'Name' with a text box containing 'Last 5', 'Collection' with a checked checkbox for 'Field', and two dropdown menus, one for 'Events Query' and one for 'Result Set'.

2. In the **Name** text box, specify a [name for the node](#). For some nodes, the name that you specify also appears as the node label in the policy diagram.
3. [Define any input values](#) for the node.
4. Configure additional settings. The specific properties that must be configured are different for each type of node.

About Node Names and Labels

Node Names

You can use the **Name** text box in the **Properties** window for a node to specify a name for the node.

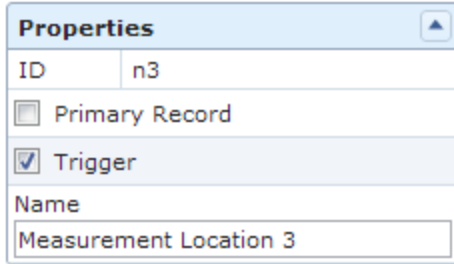
For [Condition](#) and [Action](#) nodes, the default name of each node is *<Node Type>*, where *<Node Type>* is the type of node. For example, if you add a Create Recommendation node to the policy diagram, the name of the node that appears in the corresponding **Properties** window will be *Create Recommendation*.

For [Input](#) and [Calculation](#) nodes, the default name of each of node is *<Node Type><#>*, where *<#>* is one value greater than the number of nodes that have already been added to the current policy diagram (even if they have been deleted). For example, if the policy diagram contains two nodes, when you add a new Measurement Location node, the name will be *Measurement Location 3*.

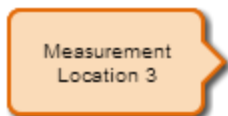
Node names for Input nodes must be unique.

Node Labels

For [Input](#) and [Action](#) nodes, the name that you specify in the **Properties** window for a node also appears as the node label in the policy diagram. For example, consider the following image of the **Properties** window for a Measurement Location node.

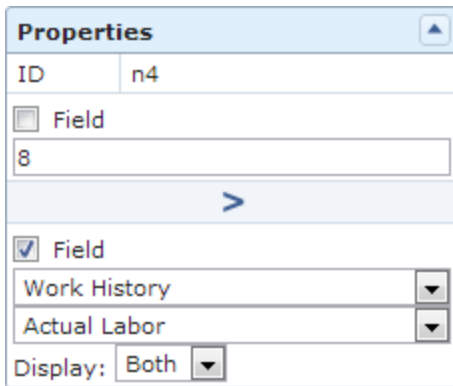


In the policy diagram, the label that appears on the Measurement Location node corresponds to the name in the **Properties** window.



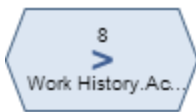
For [Calculation](#) and [Condition](#) nodes, however, the node name does not appear as the node label in the policy diagram. Instead, with the exception of the [Case node](#), these nodes have two labels which identify the first and second input values of the node. These labels appear above and below the symbol that represents the node type, respectively. For example, consider the following image of the **Properties** window for a *Greater Than* Calculation node, where:

- The first input value is 8.
- The second input value is defined by a predecessor node, where the selected node is *Work History* and the selected field is *Actual Labor*.



The Properties window is divided into two sections. The top section has an unchecked checkbox labeled 'Field' and a text input field containing the number '8'. Below this is a blue arrow pointing right. The bottom section has a checked checkbox labeled 'Field', a dropdown menu with 'Work History' selected, another dropdown menu with 'Actual Labor' selected, and a 'Display:' label with a dropdown menu set to 'Both'.

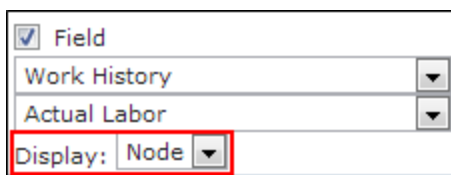
In the policy diagram, each label that appears on the Condition node identifies the input values that are defined in each section of the **Properties** window, as shown in the following image.



For an input values [defined by predecessor nodes](#), a **Display** list will appear in the corresponding section of the **Properties** window for Calculation and Condition nodes. You can use the **Display** list to specify how the corresponding label will appear. Each **Display** list contains the following options:

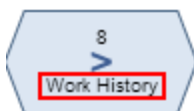
- **Node:** Specifies that the label in the policy diagram should contain the name of the *node* that is selected in the corresponding section of the **Properties** window.

Properties Window



A close-up of the Properties window's 'Field' section. The 'Field' checkbox is checked. The dropdown menu is set to 'Work History' and the second dropdown menu is set to 'Actual Labor'. The 'Display:' dropdown menu is highlighted with a red box and set to 'Node'.

Node label



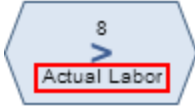
- **Field:** Specifies that the label in the policy diagram should contain the caption of the *field* that is selected in the corresponding section of the **Properties** window.

Properties Window

About Node Names and Labels

<input checked="" type="checkbox"/> Field	
Work History	▼
Actual Labor	▼
Display: Field	▼

Node label

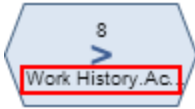


- **Both:** Specifies that the label in the policy diagram should contain *both* the name of the selected node and the caption of the selected field (separated by a period).

Properties Window

<input checked="" type="checkbox"/> Field	
Work History	▼
Actual Labor	▼
Display: Both	▼

Node label



Defining Input Values

For every node that requires an input, you will define the input value or collection using the **Field** check box and subsequent options in the **Properties** window of the node. The input can be a static value that you type directly into the **Properties** window, or the input can be a value or collection that is the output of a predecessor node.

To specify a static, user-defined input value:

1. On the **Properties** window of the node for which you want to define an input value, verify that the **Field** check box is cleared.
2. For nodes that do not require a logical input (i.e., *Yes* or *No*), in the text box, type the value that you want to use as an input.

The following image shows an example of the static, user-defined input value 5.

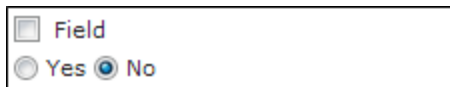


A screenshot of a software interface showing a 'Field' checkbox that is unchecked. Below it is a text input field containing the number '5'.

-or-

For nodes that require a logical input (i.e., *Yes* or *No*), select the radio button that corresponds to the value that you want to use as an input.

The following image shows an example of a static input value that is defined using a radio button, where the selected input value is *No*.

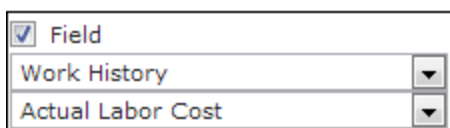


A screenshot of a software interface showing a 'Field' checkbox that is unchecked. Below it are two radio buttons labeled 'Yes' and 'No'. The 'No' radio button is selected.

To specify an input value or collection that is defined by a predecessor node:

1. On the **Properties** window of the node for which you want to define an input value, select the **Field** check box.
2. In the first list that appears, select the node that is associated with the value or collection that you want to use as the input. This list contains the names of the nodes that are [connected](#) to the selected node.
3. In the second list that appears, select the field that represents the value or collection that you want to use as the input. This list contains all of the output options that are associated with the node that you selected in the first list.

The following image shows an example of an input value that is defined by a predecessor node.



A screenshot of a software interface showing a 'Field' checkbox that is checked. Below it are two dropdown menus. The first dropdown menu is open and shows 'Work History' selected. The second dropdown menu is also open and shows 'Actual Labor Cost' selected.

Defining Input Values

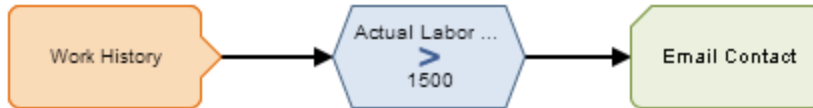
In this image, you can see that:

- The input *node* is *Work History*.
- The input *value* comes from the value in the *Actual Labor Cost* field in the Work History record (the specific Work History record will be defined by a policy instance).

[Click here](#) for an example of defining input values.

Input Values Example

To illustrate the different ways to [define input values](#), consider the following nodes and connection.



In this example, the *Work History* Entity node is connected to a *Greater Than* Condition node. The **Properties** window for the Condition node, therefore, allows you to select the *Work History* family and one of its fields. As shown in the following image:

- The **Field** check box above the greater than symbol (>) is *selected*.
- The field *Actual Labor Cost* is selected as the first input for the Greater Than node.
- The **Field** check box below the greater than symbol (>) is *cleared*.
- The value *1500* is defined as the second input.

Together, the *Work History* Entity node and the Condition node to which it is connected indicate that an email message should be sent when a *Work History* record has a value greater than 1500 in the *Actual Labor Cost* field.

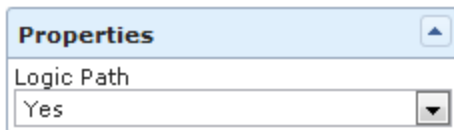
Note: You must use [policy instances](#) to identify the specific record(s) whose values you want to use in the policy. For the example provided in this topic, you would need to select a specific *Work History* record for each policy instance that is associated with the policy.

About Connection Properties

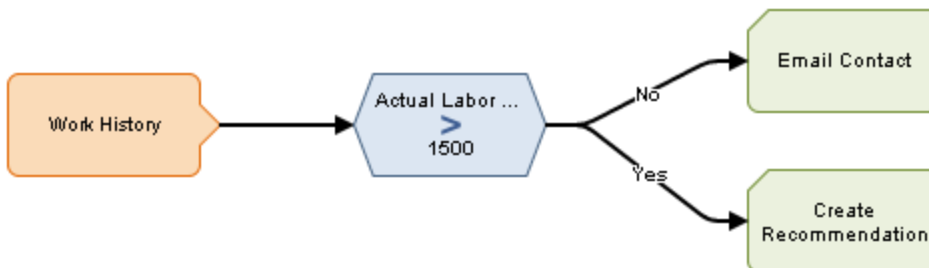
For connections that *start* at a Condition or [Logic](#) node, a **Properties** window appears when you select the connection in the policy diagram. You can use the **Properties** window to define logic that determines which path should be followed depending upon whether the condition is true or false. Specifically:

- If you set the connection property to *Yes*, the path between the predecessor node and the successor node will be followed when the logical result of the predecessor node is *true*.
- If you set the property to *No*, the path between the predecessor node and the successor node will be followed when the logical result of the predecessor node is *false*.

For example, the following image displays an example of the **Properties** window that appears when you select a connection between a Greater Than node and a Create Recommendation node:



Consider the following diagram.



Using this example, you can see that:

- The Condition node evaluates the value in the *Actual Labor Cost* field in Work History records to determine if it is greater than 1500.
- The Condition node is connected to the following successor nodes:
 - Email Contact (through the *No* connection)
 - Create Recommendation (through the *Yes* connection)

Based on the connection properties, you can determine that the following logic will be applied when the policy is executed:

- If the Actual Labor Cost is not greater than 1500, an email message will be sent. This logic is determined by the *No* property of the connection to the Email Contact node.

- If the Actual Labor Cost is greater than 1500, a Policy Recommendation record will be created. This logic is determined by the Yes property of the connection to the Create Recommendation node.

Note: If logic is not defined for a connection, a value of Yes is assumed but does not appear on the diagram.

Copying and Pasting Nodes and Connections

You can copy one or more nodes on the diagram canvas and paste them to another location on the canvas. Connections will be copied only if you select the connection and both nodes that it connects.

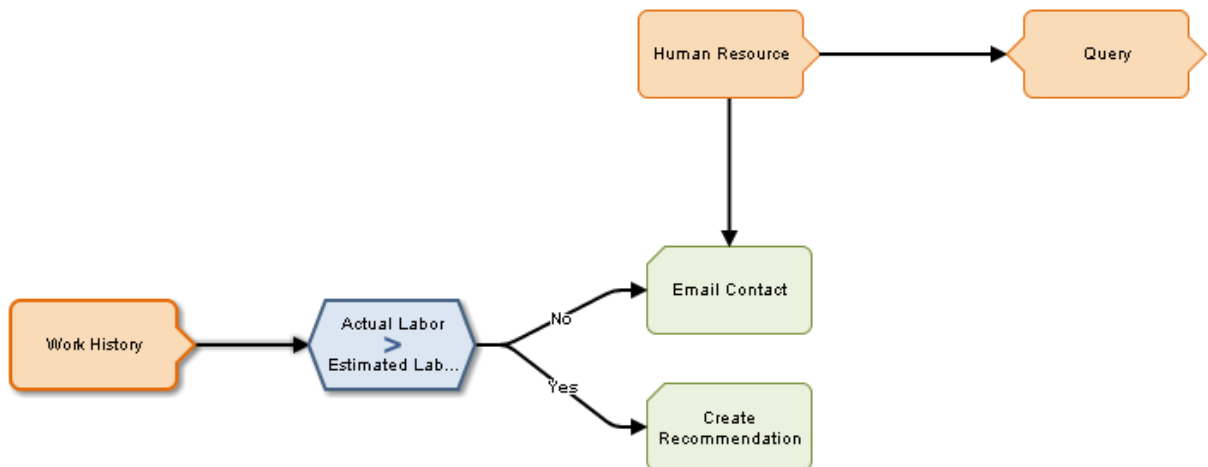
Note: After you copy a node that requires a unique name (e.g., Point Value node), you must change the name of the copied node before you can save the policy.

To copy and paste nodes:

1. On the [Policy page](#), on the diagram canvas, press the Ctrl key, and select all the nodes and connections that you want to copy.

The selected nodes and connections are outlined in a darker color. In the following image, you can see that the following items have been selected:

- *Work History* node.
- *Actual Labor > Estimated Labor* node.
- The connection between them.

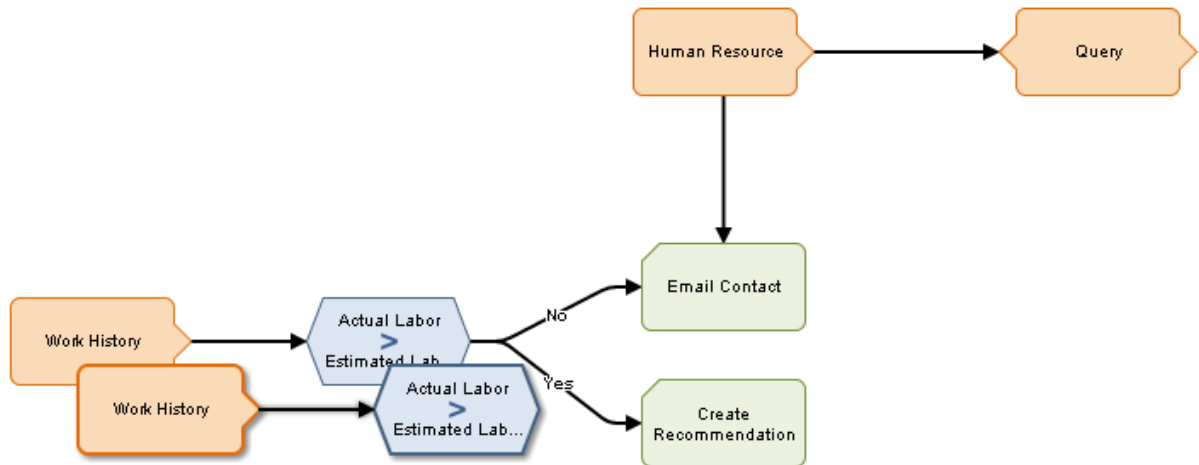


2. On the **Policy Actions** toolbar, click the  button.

The selected nodes and connections are copied.

3. On the **Policy Actions** tab, click the  button.

Copies of the selected nodes and connections are pasted to the diagram canvas and are selected automatically.



Note: If you copy a node that contains a mapped field value from another node, the mapped field value will not be copied unless you also copy the node from which the values are mapped and the connector node between the two nodes.

4. Drag the pasted nodes and connections to the desired location.

About Input Nodes

Input nodes represent items that are evaluated in order to determine if an action is required. You can add several types of Input nodes to the policy diagram using the [Inputs toolbar](#). After you add an Input node, you can define its properties using the **Properties** window. All Input nodes are shaded orange in the policy diagram. The following Input nodes are available:

- [Entity](#)
- [Query](#)
- [Point Value](#)
- [Measurement Location](#)
- [AMS Asset](#)
- [User](#)
- [Constant](#)
- [Health Indicator](#)
- [Xi Tag](#)
- [GE Tag](#)

With the exception of Query and Constant nodes, you must use [policy instances](#) to assign specific records to each Input node in the policy. For a Point Value node, you can also use policy instances to assign constant values to the node.

Note: You must have the appropriate licenses and family-level privileges to view and use nodes that are associated with specific families. For example, to view an AMS Asset node, the AMS Analytics license must be active and you must have View permissions on the AMS Asset family. Otherwise, the button that creates the AMS Asset node will be disabled.

About the Trigger Check Box

The **Trigger** check box appears on the **Properties** window for all Input nodes *except* Query, Constant, and Point Value nodes. You can use this check box to specify whether or not the policy should be executed based upon changes in the record that is associated with that *specific* node. This check box is selected by default.

Note: This setting only affects policies that are [configured to be executed when records belonging to the policy instance are updated](#).

For example, suppose you have a policy that contains an Entity node that represents an Equipment record, but this particular Entity node does *not* influence any of the logic in the policy that determines if action is needed. Rather, other Entity nodes (e.g., representing Measurement Location records that are linked to the Equipment record) influence the logic in the policy that determines if action is needed. In this case, even though the policy is [configured to be executed when records belonging to the policy instance are updated](#), you might want to specify that updates to the associated Equipment record *not* result in the execution of the policy. You can do this using the **Trigger** check box on the **Properties** window for the node.

When the **Trigger** check box is:

- **Selected:** The policy will be executed any time that the record that is associated with this node is updated. A node for which this check box is selected looks like this in the policy diagram:



- **Cleared:** The policy will *not* be executed any time that the record that is associated with this node is updated. A node for which this check box is cleared looks like this in the policy diagram:



Entity Nodes

An *Entity node* is an [Input node](#) that represents any Meridium APM entity family. You can use the Entity node to access information that is stored in a record belonging to any Meridium APM entity family. The following image shows what an Entity node looks like.



The output of an Entity node is any field in the family that you select in the **Family ID** list on the **Properties** window for the Entity node.

Properties of the Entity Node

The following image shows what the **Properties** window looks like for an Entity node after you add the node to the policy diagram.

 A screenshot of the "Properties" window for an Entity node. The window has a title bar "Properties" with a close button. It contains several fields:

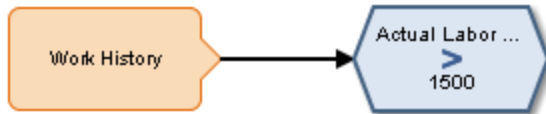
- ID: n21
- Primary Record:
- Trigger:
- Name: Entity 0
- Family ID: select... (dropdown menu)

The **Properties** window for an Entity node contains the items that are described in the following table.

Item	Description	Notes
Primary Record check box	Specifies whether or not the node represents a primary record .	None
Trigger check box	Specifies whether or not changes to the record that is associated with the node will result in policy execution .	None
Family ID	Specifies the family that the Entity node represents.	The Family ID list contains all of the Meridium APM entity families from which you can access record information. This value is required.

Example of the Entity Node

Entity nodes are often the starting point in policy diagrams because they provide successor nodes with fields to evaluate. For example, consider the following nodes and connection.



In this example, the *Work History* Entity node provides the Condition node with a field, *Actual Labor Cost*, to evaluate.

[Click here](#) for an example of an Entity node used within a complete policy diagram.

Query Nodes

A *Query node* is an [Input node](#) that represents a query that is stored in the Meridium APM Catalog. You can use a Query node to access a specific table that is defined by the specified query. The specified query will run each time the policy is executed, and you can use the results of the query in subsequent calculations. The following image shows what a Query node looks like.



If the specified query contains prompts, for each prompt that is included in the query, you must identify the values that should be provided to the prompts. To do so, you must connect the Query node to a predecessor node that represents the desired value.

A Query node generates the following outputs:

- *Result Set*, which represents the full results of the specified query. This option can be used only with successor nodes that are capable of handling collections.
- Any *value* in the top row of the specified query.

Note: A Query node cannot be the first node in a policy diagram.

Properties of the Query Node

The following image shows an example of what the **Properties** window looks like for a Query node.

Properties	
ID	n1
Name	Query
Query Path	Public\Meridium\Modules\Policy M ...
Query: Asset Key	<input checked="" type="checkbox"/> Field
	Equipment
	Entity Key

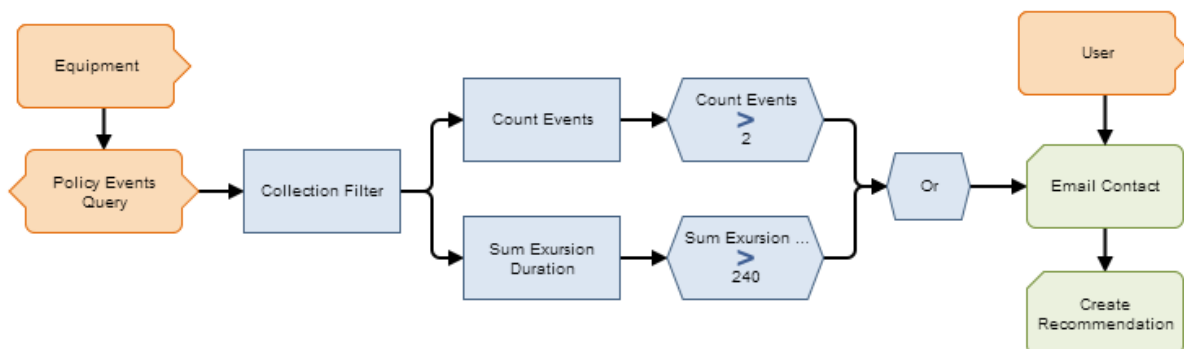
The **Properties** window for a Query node contains the items that are described in the following table.

Item	Description	Notes
Query Path	Specifies the path to the query that will run when the policy is executed.	You can type the path manually, or you can browse to it by clicking the <input type="text" value="..."/> button.
Query section	Provides values to any query prompts.	<p>One Query section appears for each prompt in the selected query. The label that appears after <i>Query:</i> identifies the prompt caption.</p> <p>The <i>Entity Key</i> and <i>Content GUID</i> system field options are also available for you to select as a prompt. If you select a predecessor node that is one of the following node types: Entity, Measurement Location, Health Indicator, AMS Asset, Xi Tag, or User.</p> <p>You will use the Field check box and subsequent options to define the value in this section.</p>

Example of the Query Node

The following example illustrates how you can use the Query node with other nodes in the policy diagram to access and analyze [Policy Event records](#) using the [Events Query](#) in the baseline Meridium APM Catalog.

Consider the following nodes and connections.



In this example, the *Policy Events Query* node is connected to the *Equipment* Entity node and to a [Collection Filter node](#), which is then connected to various additional Condition, Calculation, and Action nodes.

As shown in the following image, you can use the **Properties** window for the Query node to specify the query whose results you want to use in the policy diagram. You can also

see that the specified query contains a prompt, *Asset Key*, and that a system field from the Equipment node, *Entity Key*, is used to supply values to the prompt.

When the Query node is connected to a successor node, in the **Properties** window of the successor node, you can select the *Result Set* option to evaluate the full query results.

In this example, the Collection Filter node is used to filter the query results. As you can see in the following image of the **Properties** window for the Collection Filter node, the query results are filtered to include only the policy events that lasted longer than 10 seconds, occurred within the last month, and have the name *IOW Excursion*.

Column	Operator	Condition
Duration (seconds)	>	Field 10
Start Time	>	Field now-1 month
Name	=	Field IOW Excursion

The filtered query results are used by subsequent Calculation and Condition nodes to indicate that certain actions should be triggered when specified conditions are met.

Query Nodes

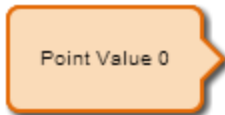
Specifically, the remaining nodes in this policy work together to send an email message and to create a recommendation if the filtered query contains more than two results or if the total duration of the query results exceeds 240 seconds.

Point Value Nodes

A *Point Value node* is an [Input node](#) that represents a specific value that you can specify using a policy instance. You can use the Point Value node to represent:

- Constant values that are used more than once in the policy diagram but are not the same value for each policy instance.
- Values from fields in different families for each policy instance.

The following image shows what a Point Value node looks like.



The output of a Point Value node is the value that you specify in a [policy instance](#).

Properties of the Point Value Node

The following image shows what the **Properties** window looks like for a Point Value node after you add the node to the policy diagram.

The **Properties** window for a Point Value node contains the items that are described in the following table.

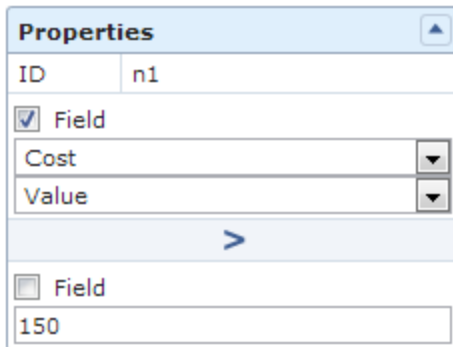
Item	Description	Notes
Trigger check box	Specifies whether or not changes to the record that is associated with the node will result in policy execution .	None
Data Type list	Specifies the type of data that the node represents.	This field is not required. However, selecting a value in this field can help catch errors and facilitate policy execution.

Example of the Point Value Node

The following example illustrates how you can use the Point Value node to represent values from fields in different families for each policy instance. Consider the following nodes and connection.



In this example, the value represented by the Point Value node is evaluated to determine whether or not it is greater than 150. As shown in the following image, you can use the **Properties** window for the *Greater Than Condition* node to select the value represented by the *Cost* Point Value node as the first input value.

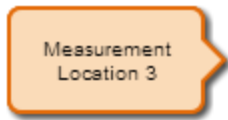


The particular record and field containing the cost value are defined by the policy instances that are associated with this policy. For example, you might create two policy instances, where:

- One policy instance maps the *Cost* field from the *Action* family to the *Cost* Point Value node.
- The other policy instance maps the *Total Cost* field from the *Work History* family to the same *Cost* Point Value node.

Measurement Location Nodes

A *Measurement Location node* is an [Input node](#) that represents the Measurement Location family. You can use the Measurement Location node to access records in the Measurement Location family and information about related readings. The following image shows what a Measurement Location node looks like.

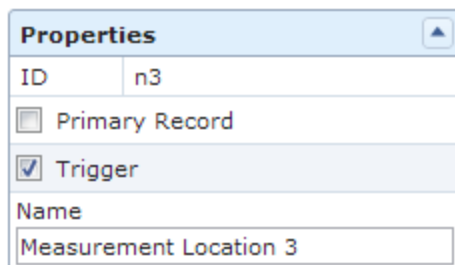


A Measurement Location node generates the following outputs:

- *Readings*, which represents the collection of all measurement readings associated with a specified Measurement Location record. This output contains a value and timestamp for each reading in the collection and can be used only with successor nodes capable of handling collections.
- *Anyfield* in the Measurement Location family.

Properties of the Measurement Location Node

The following image shows what the **Properties** window looks like for a Measurement Location node after you add the node to the policy diagram.

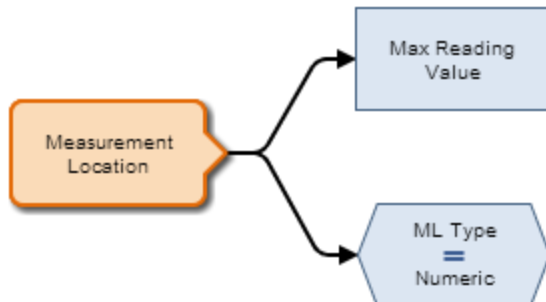


The **Properties** window for a Measurement Location node contains the items that are described in the following table.

Item	Description
Primary Record check box	Specifies whether or not the node represents a primary record .
Trigger check box	Specifies whether or not changes to the record that is associated with the node will result in policy execution .

Example of the Measurement Location Node

The following example illustrates how you can use the Measurement Location node with successor nodes to evaluate fields in a Measurement Location record and the collection of readings associated with a Measurement Location record. Consider the following nodes and connections.



In this example, the Measurement Location node is connected to a *Max* Calculation node and an *Equal* Condition node.

As shown in the following image, you can use the **Properties** window for the Max node to select the Measurement Location node and the *Readings* option. When this option is selected, the Max node evaluates the readings that are associated with the Measurement Location record and returns the reading with the highest value.

Properties	
ID	n22
Name	Max Reading Value
Collection	
<input checked="" type="checkbox"/>	Field
	Measurement Location
	Readings
Collection Column	
	Value

You can also select a *field* from the Measurement Location family as an input for a successor node. As shown in the following image, you can use the **Properties** window for the Equal node to select the *ML Type* field in the Measurement Location family. This node evaluates whether or not the ML Type field contains the exact value *Numeric*.

Measurement Location Nodes

Properties	
ID	n20
<input checked="" type="checkbox"/> Field	
Measurement Location	▼
ML Type	▼
Display:	Field ▼
=	
<input type="checkbox"/> Field	
Numeric	

[Click here](#) for an example of a Measurement Location node used within a complete policy diagram.

AMS Asset Nodes

An *AMS Asset node* is an [Input node](#) that represents the AMS Asset family. You can use the AMS Asset node to access records in the AMS Asset family and information about related alerts. The following image shows what an AMS Asset node looks like.



An AMS Asset node generates the following outputs:

- *Has Active Alerts*, which represents the value of *yes* or *no* depending on whether or not the AMS Asset has any active alerts (i.e. the AMS Asset record is linked to an AMS Asset Alert record with the value *True* in the Active field).
- *Anyfield* in the AMS Asset family.

Properties of the AMS Asset Node

The following image shows what the **Properties** window looks like for an AMS Asset node after you add the node to the policy diagram.

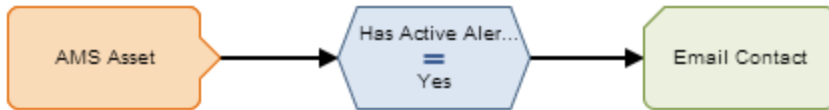
Properties	
ID	n24
<input type="checkbox"/> Primary Record	
<input checked="" type="checkbox"/> Trigger	
Name	AMS Asset 1

The **Properties** window for an AMS Asset node contains the items that are described in the following table.

Item	Description
Primary Record check box	Specifies whether or not the node represents a primary record .
Trigger check box	Specifies whether or not changes to the record that is associated with the node will result in policy execution .

Example of the AMS Asset Node

The following example illustrates how you can use the AMS Asset node with successor nodes to create conditions based upon whether or not an AMS Asset has any active alerts. Consider the following nodes and connections.



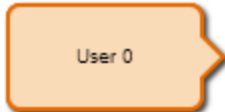
As shown in the following image, you can use the **Properties** window for the Equal node to select the AMS Asset node and the *Has Active Alerts* option. When this option is selected, the output of the AMS Asset node is either yes or no, depending on whether or not the AMS Asset is in an alert state. Together, the AMS Asset node and the Equal node indicate that an email message should be sent if the corresponding AMS Asset record has any active alerts.

You can also select a *field* from the AMS Asset family as an input for a successor node. As shown in the following image, you can use the **Properties** window for the Email Contact node to select the *Description* field in the AMS Asset family. If the Email Contact node is triggered, the email message will include the information contained in the Description field of the AMS Asset record.

[Click here](#) for an example of an AMS Asset node used within a complete policy diagram.

User Nodes

A *User node* is an [Input node](#) that represents the Security User *and* Human Resource families. You can use a User node to access user information records that are stored in either of these families. The following image shows what a User node looks.



The output of a User node is anyfield belonging to the Security User family or Human Resource family.

Properties of the User Node

The following image shows what the **Properties** window looks like for a User node after you add the node to the policy diagram.

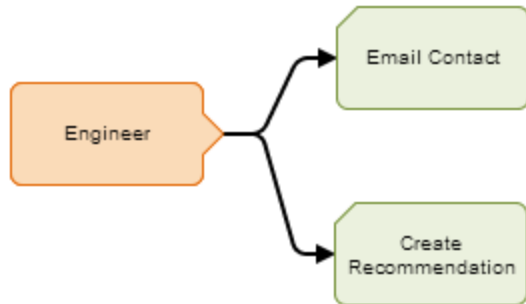
The **Properties** window for a User node contains the items that are described in the following table.

Item	Description
Primary Record check box	Specifies whether or not the node represents a primary record .
Trigger check box	Specifies whether or not changes to the record that is associated with the node will result in policy execution .

Example of the User Node

The following example illustrates how you can use the User node to access both the email address, which exists on the Human Resource family, and the User ID, which exists on the Security User family, for the same person. Consider the following nodes and connections.

User Nodes



In this example, the *Engineer* User node is connected to an Email Contact node and a Create Recommendation node.

As shown in the following image, you can use the **Properties** window for the Email Contact node to select the User node and the *Email Address* field in the Human Resource family.

Properties	
ID	n15
Name	Email Contact
To Address	
<input checked="" type="checkbox"/> Field	
Engineer	▼
Email Address	▼
Message	
<input type="checkbox"/> Field	
Example email	

Additionally, as shown in the following image, you can use the **Properties** window for the Create Recommendation node to select the User node and, in this case, a field in the Security User family, *User ID*.

Properties	
ID	n16
Name	Create Recommendation
State Assignee User ID	
<input checked="" type="checkbox"/> Field	
Engineer	▼
User ID	▼

Note that:

- Although you can select a field in the Security User *or* Human Resource family on the **Properties** window for a successor node, you can [assign only](#) Security User records to User nodes via the **Instances** tab. When you select a Human Resource

field on the **Properties** window for a successor node, the Human Resource record that is associated with the Security User record that you have assigned to the node is used for the mapping.

- Updates to the Security User *or* Human Resource record that is associated with a User node will result in the policy being executed when the [policy is configured to be executed when records belonging to the policy instance are updated](#) and the [Trigger check box](#) on the **Properties** window for the User node is selected.

[Click here](#) for an example of a User node used within a complete policy diagram.

Constant Nodes

A *Constant node* is an [Input node](#) that represents a specific value that remains constant when the policy is executed, meaning that no value is mapped to that node from a policy instance. You should use a constant node for input values that are the same for all instances and are used in multiple places in the policy diagram. The following image shows what a Constant node looks like.



The output of a Constant node is the value that you specify in the **Value** section of the node's **Properties** window.

Properties of the Constant Node

The following image shows what the **Properties** window looks like for a Constant node after you add the node to the policy diagram.

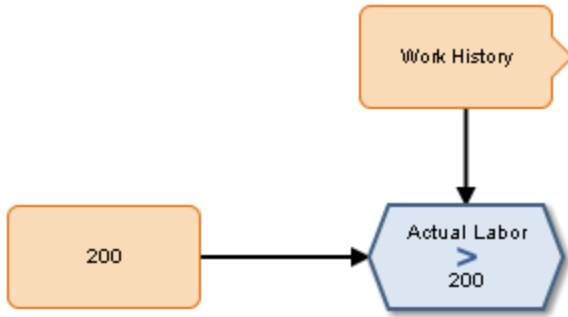
The **Properties** window for a Constant node contains the items that are described in the following table.

Item	Description	Notes
Data Type list	Specifies the type of data that the node represents.	This field is not required. However, selecting a value in this field can help catch errors and facilitate policy execution.
Value text box	Specifies the value that the node represents.	None

Example of the Constant Node

Consider the following nodes and connections.

Constant Nodes



In this example, the *200* Constant node is connected to a successor Condition node. This Condition node is also connected to another predecessor node, which is an Entity node representing the Work History family.

As shown in the following image, you can use the **Properties** window for the Condition node to select the *200* Constant node and its value, which is *200*. With these options selected, the Condition node evaluates whether or not the value in the Actual Labor field of a specified Work History record is greater than 200.

Properties	
ID	n2
<input checked="" type="checkbox"/> Field	
Work History	▼
Actual Labor	▼
Display:	Field ▼
>	
<input checked="" type="checkbox"/> Field	
200	▼
Value	▼

Health Indicator Nodes

A *Health Indicator node* is an [Input node](#) that represents the Health Indicator family. You can use the Health Indicator node to access records in the Health Indicator family and information about the readings associated with a Health Indicator's source record. The following image shows what a Health Indicator node looks like.



When you associate a policy instance with a Health Indicator node, you can assign an existing Health Indicator record to the node or [create a new Health Indicator record that will be assigned to the node](#). For Health Indicator records that are created via the **Instances** tab, you can use an [Add Value to Health Indicator node](#) to create Health Indicator Value records and link them to the new Health Indicator record.

A Health Indicator node generates the following outputs:

- *Readings*, which represents the collection of all readings associated with the Health Indicator's source record. This output contains a value and timestamp for each reading in the collection and can be used only with successor nodes capable of handling collections.
- *Anyfield* in the Health Indicator family.

Properties of the Health Indicator Node

The following image shows what the **Properties** window looks like for a Health Indicator node after you add the node to the policy diagram.

Properties	
ID	n27
<input type="checkbox"/> Primary Record	
<input checked="" type="checkbox"/> Trigger	
Name	Health Indicator 1

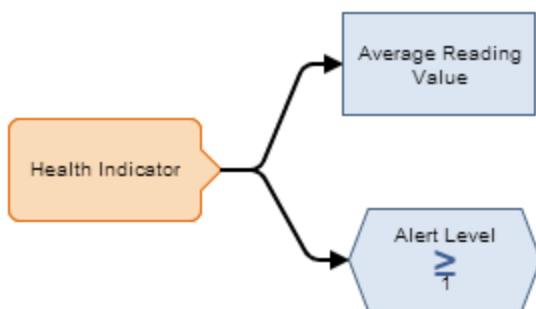
The **Properties** window for a Health Indicator node contains the items that are described in the following table.

Item	Description
Primary Record check box	Specifies whether or not the node represents a primary record .

Item	Description
Trigger check box	Specifies whether or not changes to the record that is associated with the node will result in policy execution .

Example of the Health Indicator Node

The following example illustrates how you can use the Health Indicator node with successor nodes to evaluate fields in a Health Indicator record and the collection of readings associated with a Health Indicator record. Consider the following nodes and connections.



In this example, the Health Indicator node is connected to an *Average Calculation* node and a *Greater Than or Equal Condition* node.

As shown in the following image, you can use the **Properties** window for the Average node to select the Health Indicator node and the *Readings* option. When these options are selected, this node evaluates the readings that are associated with the Health Indicator record and returns the average reading value.

The screenshot shows a "Properties" window with the following fields and values:

- ID: n24
- Name: Average Reading Value
- Collection:
 - Field
 - Health Indicator (dropdown)
 - Readings (dropdown)
- Collection Column: Value (dropdown)

You can also select a *field* from the Health Indicator family as an input for a successor node. As shown in the following image, you can use the **Properties** window for the Greater than or Equal node to select the *Alert Level* field in the Health Indicator family. This node, therefore, evaluates whether or not the value in the Alert Level field of the

corresponding Health Indicator record is greater than or equal to 1, which would indicate that the Health Indicator record meets the *Warning* or *Alert* status criteria.

The screenshot shows a 'Properties' dialog box with the following elements:

- ID:** n30
- Field**
- Health Indicator:** (dropdown menu)
- Alert Level:** (dropdown menu)
- Display:** Field (dropdown menu)
- Operator:** \geq
- Field**
- Value:** 1

[Click here](#) for an example of a Health Indicator node used within a complete policy diagram.

Xi Tag Nodes

An *Xi Tag node* is an [Input node](#) that represents the Xi Tag family. You can use the Xi Tag node to access records in the Xi Tag family and the related reading data that is accessed directly from the process historian. The following image shows what an Xi Tag node looks like.

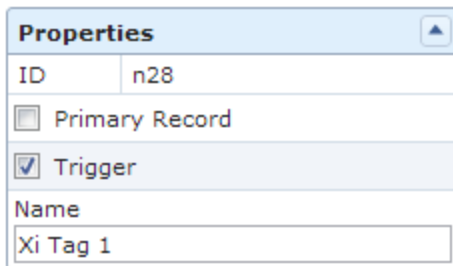


An Xi Tag node generates the following outputs:

- *DA Reading Value*, which represents the most current reading available in the process historian.
- *DA Readings Timestamp*, which represents the date and time that the most current reading was recorded in the process historian.
- *HDA Readings*, which represents the collection of historical readings taken over a period of time from the process historian. This option contains a value and timestamp for each reading in the collection and can be used only with successor nodes capable of handling collections.
- Anyfield in the Xi Tag family.

Properties of the Xi Tag node

The following image shows what the **Properties** window looks like for an Xi Tag node after you add the node to the policy diagram.

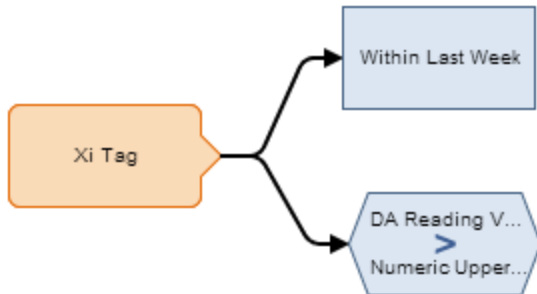


The **Properties** window for an Xi Tag node contains the items that are described in the following table.

Item	Description
Primary Record check box	Specifies whether or not the node represents a primary record .
Trigger check box	Specifies whether or not changes to the record that is associated with the node will result in policy execution .

Example of the Xi Tag Node

The following example illustrates how you can use the Xi Tag node with successor nodes to evaluate various different types of data related to the Xi Tag family. Consider the following nodes and connections.



In this example, the Xi Tag node is connected to a Collection Filter node and to a *Greater Than* Condition node.

As shown in the following image, you can use the **Properties** window for the Collection Filter node to select the Xi Tag node and the *HDA Readings* option. When this option is selected, the Collection Filter node evaluates the historical readings associated with the Xi Tag family and filters the collection to only the readings that have a timestamp within the last seven days.

Properties			
ID	n2		
Name	Within Last Week		
Collection	<input checked="" type="checkbox"/> Field Xi Tag HDA Readings		
+ Filter			
Column		Condition	
Timestamp	>	<input type="checkbox"/> Field now - 7d	X

You can also use the Xi Tag node to evaluate the *most recent reading value* associated with an Xi Tag record or values contained in a *field* of an Xi Tag record. As shown in the following image, you can use the **Properties** window for the Greater Than node to evaluate whether or not the *DA Reading Value* is greater than the value defined in the *Numeric Upper Level 2* field of the Xi Tag record.

Xi Tag Nodes

Properties	
ID	n35
<input checked="" type="checkbox"/> Field	
Xi Tag	▼
DA Reading Value	▼
Display:	Field ▼
>	
<input checked="" type="checkbox"/> Field	
Xi Tag	▼
Numeric Upper Level 2	▼
Display:	Field ▼

[Click here](#) for an example of an Xi Tag node used within a complete policy diagram.

GE Tag Nodes

A *GE Tag node* is an [Input node](#) that represents the GE Tag family. You can use the GE Tag node to access records in the GE Tag family and information about related events. The following image shows what a GE Tag node looks like.



A GE Tag node generates the following outputs:

- *GE Tag Events*, which represents the collection of GE Tag Event records that are associated with a specified GE Tag record. This output contains the start time, severity level, and type of each event in the collection and can be used only with successor nodes capable of handling collections.
- Any *field* in the GE Tag family, as well as the *Entity Key* and *Content GUID* system fields in the GE Tag family.

Properties of the GE Tag Node

The following image shows what the **Properties** window looks like for a GE Tag node after you add the node to the policy diagram.

The **Properties** window for a GE Tag node contains the items that are described in the following table.

Item	Description
Primary Record check box	Specifies whether or not the node represents a primary record .
Trigger check box	Specifies whether or not changes to the record that is associated with the node will result in policy execution .

Example of the GE Tag Node

The following example illustrates how you can use the GE Tag node with successor nodes to evaluate a related GE Tag Event record. Consider the following nodes and connections.



In this example, the GE Tag node is connected to a Last node and to a *Greater Than* Condition node. As shown in the following image, you can use the **Properties** window for the Last node to select the GE Tag node and the *GE Tag Events* option. When this option is selected, the Last node evaluates the events that are associated with the GE Tag record and returns the most recent event.

Properties	
ID	n1
Name	Last
Collection	
<input checked="" type="checkbox"/> Field	
GE Tag	▼
GE Tag Events	▼

You can then use subsequent nodes to evaluate information related to the most recent event. As shown in the following image, you can use the Greater Than node to evaluate whether or not the most recent event has a severity greater than 0.

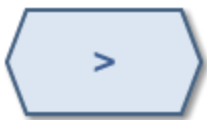
Properties	
ID	n2
<input checked="" type="checkbox"/> Field	
Last	▼
Severity Level	▼
Display:	Field ▼
>	
<input type="checkbox"/> Field	
0	

Comparison Nodes

Comparison nodes are Condition nodes that you can use to compare two input values using the comparison operator that corresponds to the name of the node. The following comparison nodes are available:

- Equal
- Not Equal
- Greater Than
- Greater Than or Equal
- Less Than
- Less Than or Equal

You can use the **Conditions** toolbar to add comparison nodes to the policy diagram. Comparison nodes are shaded blue in the policy diagram. The following image shows what a Greater Than (>) comparison node looks like after you add it to the policy diagram.



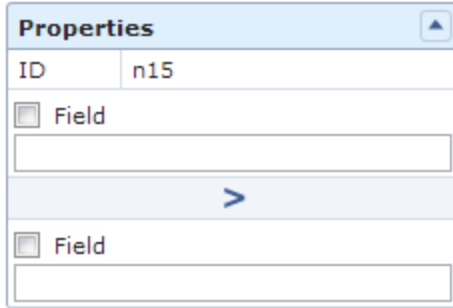
Each comparison node requires two inputs, which must be single values. For Equal and Not Equal nodes, inputs can be any type of data. For the remaining comparison nodes, inputs must be numeric or time-based values.

The output of a comparison node is the logical result of the comparison (i.e., *yes* or *no*). The output of a comparison node can only be used as an input to Case or Logic nodes. For all other successor nodes, the output is used by the *connection* to the successor node in order to determine if the successor node will be executed.

You can use the [Properties window for a connection starting at a comparison node](#) to specify a logic path for the connection. If you do not specify a logic path, a value of *Yes* is assumed. The Meridium APM system will execute only the branches of a policy diagram where the logical result of the comparison matches the logic path defined for the corresponding connection.

Properties of Comparison Nodes

The following image shows what the **Properties** window looks like for a Greater Than node after you add the node to the policy diagram.

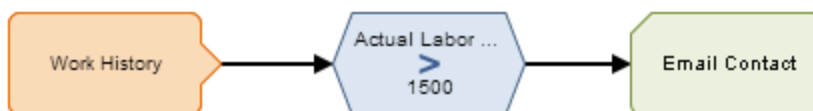


The **Properties** window for each comparison node contains the items that are described in the following table.

Item	Description	Notes
First value	The input value that will be compared to the second input value.	You will use the Field check box and subsequent options to define the value in this section.
Operator symbol	The symbol that corresponds with the comparison operation that is performed by the node.	None
Second value	The input value that will be compared to the first input value.	You will use the Field check box and subsequent options to define the value in this section.
Display list	The label that appears on the node in the policy diagram.	This list does not appear if you typed a constant in the corresponding section.

Example of a Comparison Node

The following example illustrates how you can use a comparison node to compare the value in a field of a predecessor node to a constant value. Consider the following nodes and connections.



In this example, the Greater Than node and the Work History node to which it is connected indicate that an email message should be sent when a Work History record is saved with a value greater than 1500 in the Actual Labor Cost field. The following image shows what the **Properties** window looks like for the Greater Than node.

Comparison Nodes

The screenshot shows a 'Properties' dialog box with the following fields and values:

ID	n26
<input checked="" type="checkbox"/> Field	
Work History	▼
Actual Labor Cost	▼
Display:	Field ▼
>	
<input type="checkbox"/> Field	
	1500

In the example diagram, the connection property between the *Greater Than* Condition node and the Email Contact node is not defined. Therefore, a value of *Yes* is assumed but does not appear on the diagram. Further, because this diagram does not contain a logic path to follow if the result of the comparison is *false*, the policy execution will continue (i.e. send an email message) only if the logical result of the Greater Than node is *true*.

[Click here](#) for an example of a comparison node used in a policy diagram with two separate logic paths.

Case Nodes

A *Case node* is a Condition node that you can use in the policy diagram to set up scenarios in which the output values of the node should be changed automatically based on specific input values. Throughout this documentation, we refer to each defined scenario as a *Case*. Each Case within the Case node has an input value and one or more output values that the Meridium APM system will use if the value in a defined input field matches the input value of the Case.

There are two types of Cases in the Case node, which we refer to as the *If Case* and *Else Case* throughout this documentation:

- **If Case:** The type of Case that is executed if the value in a defined input field matches a specific input value that you specify. You can define one or more If Cases for a given Case node.
- **Else Case:** The Case that is executed by default if the value in a defined input field *does not match* a specific input value that you specified in an If Case. There is only one Else Case for a given Case node.

You can use the Case node to:

- [Change the output value of a node from one value to another.](#)
- [Combine policy logic after a Condition node in the policy diagram.](#)

You can use the **Conditions** toolbar to add a Case node to the policy diagram. Case nodes are shaded blue in the policy diagram. The following image shows what the Case node looks like.



The input of a Case node must be a single value or the logical result of a [comparison node](#). The output of a Case node is the value that you define in the **Value** column corresponding to the output option that you select in a successor node. Outputs may be single values or collections.

Properties of the Case Node

The following image shows what the **Properties** window looks like for a Case node after you add the node to the policy diagram.

Properties

ID: n1

Name: Case

Input

Field

+ Case

if Input =

Mapping	Value:
Output 1	<input type="checkbox"/> Field <input type="text"/>

+




else:


Mapping	Value:
Output 1	<input type="checkbox"/> Field <input type="text"/>

+

The **Properties** window for a Case node contains the items that are described in the following table.

Item	Description	Notes
Input section	Specifies the value that you want to compare to each specific Case input.	You will use the Field check box and subsequent options to define the value in this section. If you select a comparison Condition node as a predecessor node, the logical result of the condition will be used automatically as the input value.
Add Case button	Adds a new If Case to the Case node.	Each If Case contains an If Input = text box, Remove Case <input type="button" value="X"/> button, output grid, and Add Field Mapping <input type="button" value="+"/> button.

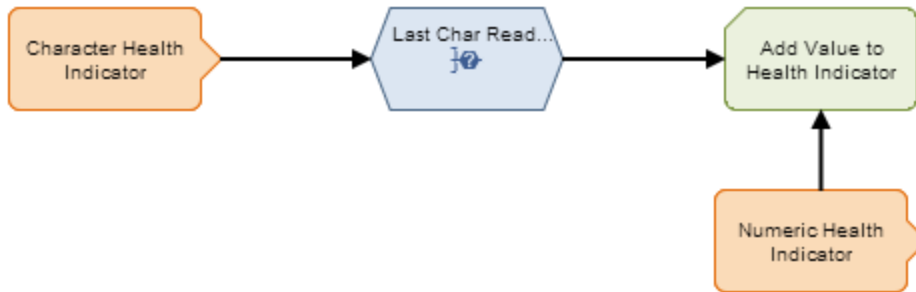
Item	Description	Notes
If Case section		
If Input = text box	Specifies the value that must match the value that is defined in the Input section in order for the Case to be executed.	None
 button	Deletes the corresponding Case from the Case node.	This button appears to the right of the If Input = text box.
Output grid	<p>Specifies the output value or collection for each Case. Each row of the output grid for each Case contains the following items:</p> <ul style="list-style-type: none"> • Mapping column: Specifies a name for each output that you define. • Value column: Specifies what the output value or collection will be if the corresponding input value matches the value defined in the Input section of a Case node. •  button: Deletes the corresponding output row from the Case node. 	<p>When changes are made to the name of an output, the name of corresponding outputs in the Mapping column of other Cases are updated automatically.</p> <p>You will use the Field check box and subsequent options to define the value in the Value column.</p> <p>When you delete an output row from one Case, corresponding output rows are automatically deleted from all Cases.</p>
 button	Adds additional rows to the output grid.	When you add an output row to one Case, corresponding output rows are added automatically to all Cases.
Else Case section		

Item	Description	Notes
Else section	Defines the Else Case for the Case node.	<p>This section contains an output grid and a  button. These items have the same functionality as the corresponding items in the If Case section.</p> <p>You cannot delete the Else Case from the Case node.</p>

Using the Case Node to Change Output Values

You can use the [Case node](#) in the policy diagram to change a node's output values so that they can be used in successor nodes to perform calculations or actions. For example, you can use the Case node to convert character-based values from a Health Indicator record reading or an AMS Asset status message to numeric values that can then be used in subsequent calculations.

Consider the following policy diagram in which reading values from a character-based Health Indicator record are converted to numeric values. The values are then added to a Health Indicator record that uses numeric record values.



The following table describes each node that is used in this example.

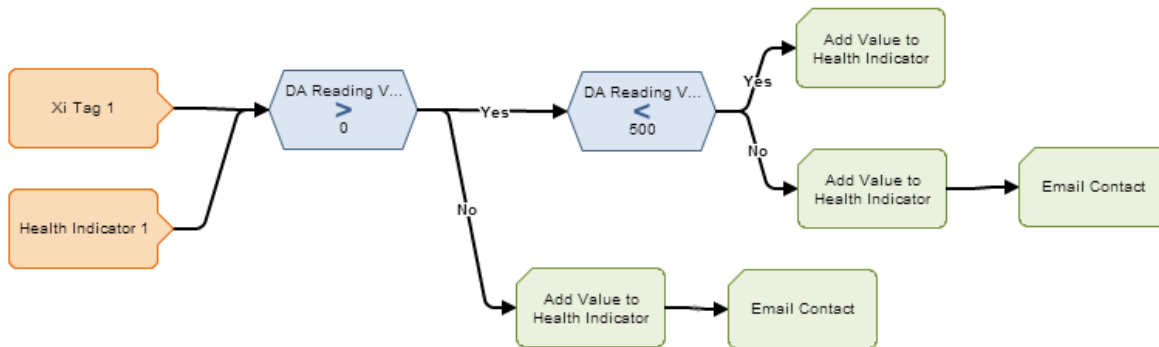
Node Type	Properties Window	Description
Health Indicator	<div style="border: 1px solid #ccc; padding: 5px;"> <p>Properties</p> <p>ID <input type="text" value="n7"/></p> <p><input type="checkbox"/> Primary Record</p> <p><input checked="" type="checkbox"/> Trigger</p> <p>Name <input type="text" value="Character Health Indicator"/></p> </div>	Represents a character-based Health Indicator record.

Node Type	Properties Window	Description
<p>Case</p>		<p>Indicates that the value in the <i>Last Char Reading Value</i> field of the <i>Character Health Indicator</i> node will be compared with defined input values to determine the output of the Case node.</p> <p>In this example, if the value in the <i>Last Char Reading Value</i> field is:</p> <ul style="list-style-type: none"> • <i>1-OK</i>, the output of the node will be <i>100</i>. • <i>2-Not OK</i>, the output will be <i>50</i>. <p>Otherwise, the output will be <i>0</i>.</p>
<p>Health Indicator</p>		<p>Represents a Health Indicator record that uses numeric values.</p>
<p>Add Value to Health Indicator</p>		<p>Indicates that the values in the fields specified in the Timestamp and Value (numeric) sections will be added to the numeric Health Indicator record that is represented by the Health Indicator node with the name <i>Numeric Health Indicator</i>.</p> <p>The field specified in the Value (numeric) section contains the numeric equivalents of the character-based readings from the Health Indicator node with the name <i>Character Health Indicator</i>.</p>

Using the Case Node to Combine Policy Logic

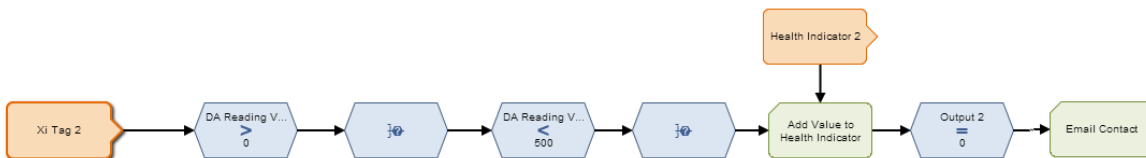
You can use the [Case node](#) to combine policy logic in situations when a Condition node's successor nodes are the same in each logic branch. Rather than creating separate logic paths with duplicate nodes, you can use a Case node to define different outputs for each logical result of the Condition. This is especially useful in large policy diagrams with multiple logic branches and duplicate nodes.

The following image shows what a policy that uses Condition nodes looks like without Case nodes in the diagram.



Notice that the *Add Value to Health Indicator* node appears three times, and the *Email Contact* node appears twice.

The following image shows what the same policy looks like when you use Case nodes in the diagram.



In this diagram, each Case node takes the place of a logic branch. Rather than splitting the diagram into separate logic paths, the Case node supplies a specified output corresponding to the logical result of the preceding Condition node. Notice that in this diagram the *Add Value to Health Indicator* node and the *Email Contact* node appear only once.

The following image shows what the **Properties** window looks like for the first Case node in the preceding image.

Properties

ID: n1

Name: Case

Input: Field

Greater Than

+ Case

if Input = Yes

Mapping	Value:
Output 1	<input type="checkbox"/> Field 50

+ else:

Mapping	Value:
Output 1	<input type="checkbox"/> Field 0

In this example, if the logical result of the Condition node named *Greater Than* is:

- Yes, the output of the node will be 50.
- No, the output will be 0.

When *Output 1* is selected as the input of a successor node, the Meridium APM system will supply the corresponding value to the successor node.

Logic Nodes

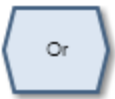
You can use *Logic nodes* to evaluate the logical results of all nodes that immediately precede the Logic node. The following Logic nodes are available:

- **And:** Evaluates whether or not the logical result of *all* immediately preceding nodes is *yes*.
- **Or:** Evaluates whether or not the logical result of *at least one* immediately preceding node is *yes*.

You can use the [Logic toolbar](#) to add Logic nodes to the policy diagram. Logic nodes are shaded blue in the policy diagram. The following image shows what an *And* node looks like:



...and this image shows what an *Or* node looks like:



In the policy diagram, Logic nodes must be preceded immediately by comparison or other Logic nodes. The logical results of these nodes will be used automatically as the input values for the Logic node.

When a Logic node is executed:

- If the Logic node's criteria is met, the logical result of the node will be *yes*.
- If the Logic node's criteria is not met, the logical result of the node will be *no*.

The following table summarizes what the result of each Logic node will be for various input combinations.

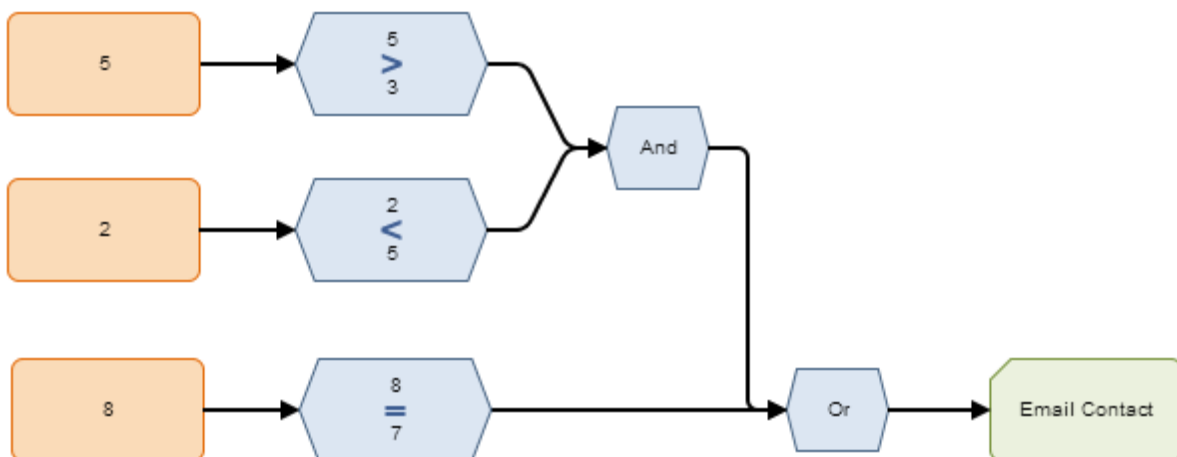
Input A	Input B	And Node	Or Node
Yes	Yes	Yes	Yes
No	No	No	No
Yes	No	No	Yes

Note: For *Or* nodes, any input value that is not *yes* is considered *no*. This means that if a preceding node is not executed or if errors occur during execution, the input from the corresponding node will be *no*.

The logical results of Logic nodes are used by *connections* to successor nodes in order to determine if the successor node will be executed. You can use the [Properties window for a connection starting at a Logic node](#) to specify a logic path for the connection. If you do not specify a logic path, a value of Yes is assumed but does not appear on the diagram. The Meridium APM system will execute only the branches of a policy diagram where the logical result of the Logic node matches the logic path defined for the corresponding connection.

Example of Logic Nodes

The following example illustrates how you can use Logic nodes to monitor policy execution. To simplify this example, we have used only constant values in the policy diagram. Consider the following nodes and connections.



In this example:

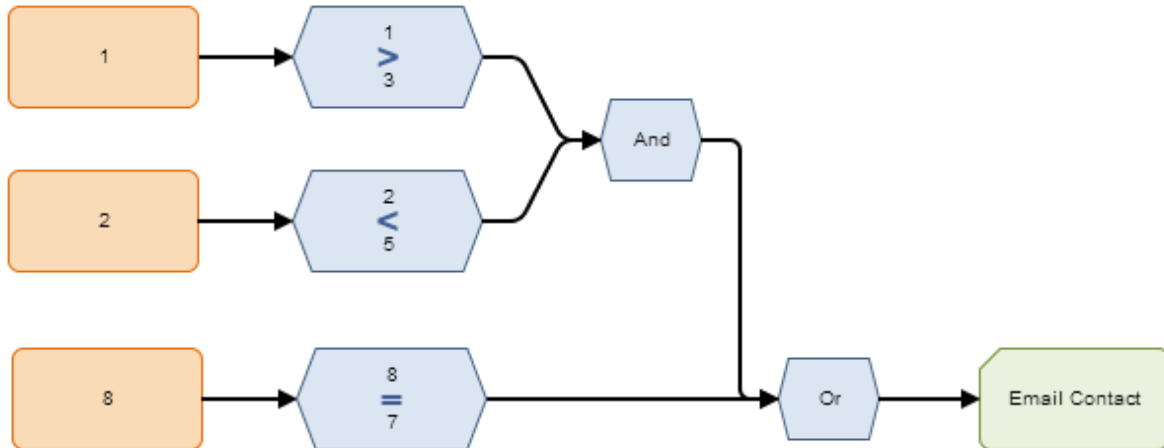
- The condition $5 > 3$ is true, which means that the result of the Greater Than node preceding the And node is *yes*.
- The condition $2 < 5$ is true, which means that the result of the Less Than node preceding the And node is *yes*.
- The condition $8 = 7$ is false, which means that the result of the Equal node preceding the Or node is *no*.
- Logic paths are not defined for the connections that start at the Logic nodes. Therefore, for each connection, a value of Yes is assumed but does not appear on the diagram.

Based on these results, when the policy is executed:

- The logical result of the And node will be *yes* because all of its inputs are *yes*.
- The logical result of the Or node will be *yes* because at least one of its inputs (i.e., the input from the And node) is *yes*.
- Policy execution will continue (i.e., an email message will be sent to a specified

contact) because the logic path for the connection starting at the Or node (i.e., yes) matches the logical result of the Or node.

Continuing with this example, suppose that the first input value in this policy changes. Consider the following nodes and connections.



In this example:

- The condition $1 > 3$ is false, which means that the result of the Greater Than node preceding the And node is *no*.
- The condition $2 < 5$ is true, which means that the result of the Less Than node preceding the And node is *yes*.
- The condition $8 = 7$ is false, which means that the result of the Equal node preceding the Or node is *no*.
- Logic paths are not defined for the connections that start at the Logic nodes. Therefore, for each connection, a value of Yes is assumed but does not appear on the diagram.

Based on these results, when the policy is executed:

- The logical result of the And node will be *no* because one of its inputs is *no*.
- The logical result of the Or node will be *no* because both of its inputs are *no*.
- Policy execution will *not* continue (i.e., an email message will not be sent) because the logic path for the connection starting at the Or node (i.e., yes) does not match the logical result of the Or node.

About Calculation Nodes

You can use *Calculation Nodes* to perform calculations on various values or collections of data that are represented by nodes in the policy diagram. You can add Calculation nodes to the policy diagram using the [Calculations toolbar](#). After you add a Calculation node, you can define its properties using the [Properties window](#). Calculation nodes are shaded blue in the policy diagram.

You can use the following Calculation nodes to perform calculations on single values:

- [Add](#)
- [Subtract](#)
- [Multiply](#)
- [Divide](#)
- [Remainder](#)
- [Exponent](#)
- [Round](#)

You can use the following Calculation nodes to perform calculations on collections of data (i.e. collections of results from a Query node or collections of readings associated with Measurement Location, Health Indicator, or Xi Tag nodes):

- [Collection Filter](#)
- [Last](#)
- [Min](#)
- [Max](#)
- [Average](#)
- [Sum](#)
- [Count](#)
- [Threshold Statistics](#)

Add, Subtract, Multiply, Divide, Remainder, and Exponent Nodes

The following [Calculation nodes](#) are also called *math nodes* because they represent basic mathematical calculations:

- **Add:** Adds one value to another value.
- **Subtract:** Subtracts one value from another value.
- **Multiply:** Multiplies one value by another value.
- **Divide:** Divides one value by another value.
- **Remainder:** Calculates the value remaining after one value is divided by another value.
- **Exponent:** Raises one value to the power of another value.

The following image shows what a math node looks like after you add it to the policy diagram. The symbol that appears in the middle of the node corresponds to the type of node (e.g., +).



Each math node has two inputs. Input requirements differ depending upon the type of math node that you are using.

- For Multiply, Divide, Remainder, and Exponent Nodes, inputs must be single, numeric values.
- For Add and Subtract nodes, inputs can be single, numeric values or certain time-based values.

The output of all math nodes is a single field, *Answer*, which represents the result of the calculation.

Properties of Math Nodes

The following image shows what the **Properties** window looks like for an Add node after you add the node to the policy diagram.

The **Properties** window for each math node contains the items that are described in the following table.

Item	Description	Notes
First value	The first input value that will be used in the calculation.	You will use the Field check box and subsequent options to define the value in this section.
Calculation symbol	The symbol that corresponds with the mathematical calculation that is performed by the node.	None
Second value	The second input value that will be used in the calculation.	You will use the Field check box and subsequent options to define the value in this section.
Display list	The label that appears on the node in the policy diagram.	This list does not appear if you typed a constant in the corresponding section.

The following table illustrates how the input values that you define for each node correspond to the mathematical calculations performed by the node. This table includes the mathematical calculation and output for each type of node when the first input value is 3 and the second input value is 2, as shown in this example of an Exponent node:

Node Type	Mathematical Calculation	Output Value
Add	$3+2$	5
Subtract	$3-2$	1
Multiply	$3*2$	6
Divide	$3/2$	1.5
Remainder	$3\%2$	1
Exponent	3^2	9

Using time-based values with Add and Subtract Nodes

In addition to operating on single, numeric input values, Add and Subtract nodes can perform calculations on certain time-based input values.

The following tables summarize the possible combinations of timestamps and time spans that you can use with Add and Subtract Nodes. *First Value* is the input value defined in the top section of the Properties window, and *Second Value* is the input value defined in the bottom section of the **Properties** window.

Add Node

First Value	Second Value	Output	Example
Timestamp	Time span	Timestamp	$1/11/2000 + 10 \text{ Days} = 1/21/2000$
Time span	Timestamp	Timestamp	$10 \text{ Days} + 1/11/2000 = 1/21/2000$
Time span	Time span	Time span	$12 \text{ Days} + 10 \text{ Days} = 22 \text{ Days}$
Timestamp	Timestamp	Invalid	N/A

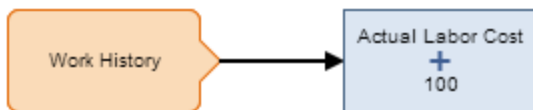
Subtract Node

First Value	Second Value	Output	Example
Timestamp	Time span	Timestamp	$1/11/2000 - 10 \text{ Days} = 1/1/2000$
Time span	Timestamp	Invalid	N/A

First Value	Second Value	Output	Example
Time span	Time span	Time span	12 Days - 10 Days = 2 Days
Timestamp	Timestamp	Time span	1/11/2000 - 1/1/2000 = 10 Days

Example of a Math Node

The following example illustrates how you can use an Add node to add a constant value to a value defined by a predecessor node. Consider the following nodes and connection.



In this example, the Add node adds 100 to the value in the *Actual Labor Cost* field of a particular *Work History* record. The following image shows what the **Properties** window looks like for the Add node.

The screenshot shows the 'Properties' window for an 'Add' node. The ID is 'n3'. The Name is 'Add 3'. The 'Field' checkbox is checked. The 'Work History' field is selected in the dropdown menu. The 'Actual Labor Cost' field is also selected in the dropdown menu. The 'Display' dropdown is set to 'Field'. A plus sign (+) is visible below the field selection. The 'Field' checkbox is unchecked, and the value '100' is entered in the text field below.

[Click here](#) for an example of math nodes used within a complete policy diagram.

Round Nodes

A *Roundnode* is a [Calculation node](#) which you can use in a policy diagram to:

- Round a value to a specific number of decimal places. This is called *decimal rounding*.
- or-
- Round a value to a specific number of significant figures. This is called *precision rounding*.

The following image shows what a Round node looks like.



When you use decimal rounding, you can specify the number of *decimal places* to which you want to round the input value. For example, if the input value is *3.51* and you indicate that you want to round to the nearest tenth (by specifying the value *1* in the **Digits** text box on the **Properties** window), the input value will be rounded to *3.5*.

When you use precision rounding, you can specify the number of *digits that you want keep* from the input value. For example, if the input value is *7,658,321* and you indicate that you want to keep the first three values (by specifying the value *3* in the **Digits** text box on the **Properties** window), the input value will be rounded to *7,660,000* (where the third digit, *5*, is rounded to *6*, and the remaining digits are changed to *0*).

The input of a Round node must be a single, integer value. The output of a Round node, *Answer*, contains the rounded value.

Properties of the Round Node

The following image shows what the **Properties** window looks like for a Round node after you add the node to the policy diagram.

Round Nodes

The screenshot shows a 'Properties' window for a 'Round' node. The 'ID' is 'n29' and the 'Name' is 'Round 4'. Under the 'Value' section, the 'Field' checkbox is checked, and there is an empty text box below it. Under the 'Digits' section, the 'Field' checkbox is also checked, with another empty text box below it. The 'Mode' is set to 'Decimal rounding'.

The **Properties** window for a Round node contains the items that are described in the following table.

Item	Description	Notes
Value section	Specifies the value that will be rounded.	You will use the Field check box and subsequent options to define the value in this section.

Item	Description	Notes
Digits section	<p>The value in this section serves a different purpose depending on the option you select in the Mode section. You can select:</p> <ul style="list-style-type: none"> • Decimal rounding: Specifies the number of <i>decimal places</i> to which you want to round the input value. • Precision rounding: Specifies the number of <i>digits that you want keep</i> from the input value. 	<p>You will use the Field check box and subsequent options to define the value in this section.</p> <p>If the list in the Mode section contains the value:</p> <ul style="list-style-type: none"> • Decimal rounding, you must type a value between 0 and 15 in this text box. • Precision rounding, you must type a value <i>greater than zero</i>. <p>Regardless of the values that you specify for the Round node, when you validate the policy logic, to keep the policy diagram easier to read, it displays only <i>two</i> decimals. You can see the actual values by accessing the Properties window for the appropriate node.</p>
Mode list	Specifies the type of rounding that the node will perform.	<p>This list contains the following options:</p> <ul style="list-style-type: none"> • Decimal rounding • Precision rounding <p>By default, the Mode list contains the value Decimal rounding and the Digits text box is empty (this is the same as typing <i>0</i> [zero]). This means that decimal values will be rounded to the nearest whole number.</p>

Example of the Round Node

Round Nodes

The following example illustrates how you can use the Round node in the policy diagram to round the value of an Xi Tag reading, represented by the *Pressure* node, to the nearest ten. Consider the following nodes and connection.



As shown in the following image, you can use the **Properties** window for the Round node to specify the type of rounding which you want to use. In this example, *precision rounding* is selected as the type of rounding and *1* is specified as the number of digits from the input value that will be kept. This means that if, for example, the input to the node is *68*, the output value will be *70*.

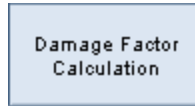
Properties	
ID	n8
Name	Round to the nearest 10
Value	
<input checked="" type="checkbox"/> Field	
Pressure	▼
DA Reading Value	▼
Digits	
<input type="checkbox"/> Field	
1	
Mode	
Precision rounding	▼

The Round node in this example could be connected to various successor nodes in order to facilitate calculations. You could, for example, connect the Round node to a Case node, and then define a specific case to be executed based on the rounded input (i.e., 10, 20, 30, and so on). Using the Round node in this way facilitates the execution of the Case node by limiting the number possible inputs.

[Click here](#) for an example of a Round node used within a complete policy diagram.

R-Script Nodes

An *R-Script node* is a [Calculation node](#) that you can use in a policy diagram to return a result that is calculated by an RScript created outside the policy using the RScript Editor.



The inputs for an R-Script node must correspond to the type of inputs expected by the RScript. The output of an R-Script node contains the value calculated by the RScript.

Properties of the R-Script Node

The following image shows what the **Properties** window looks like for an R-Script node after you add the node to the policy diagram.

Properties	
ID	n14
Name	Damage Factor Calculation
RScript Path	
<input type="checkbox"/> Field	
	Public\Meridium\Modules\Risk Ba ...
LinerBaseDF	
<input checked="" type="checkbox"/> Field	
	Inorganic Base DF Lookup ▼
	Damage Factor ▼
LiningConditionAdjustment	
<input checked="" type="checkbox"/> Field	
	Lining Condition Adjustment Look ▼
	Adjustment Multiplier ▼
OnlineMonitoringAdjustment	
<input checked="" type="checkbox"/> Field	
	RBI 581 Thinning and Lining ▼
	Online Monitoring Adjustment ▼

The **Properties** window for an R-Script node contains the items that are described in the following table.

Item	Description	Notes
RScript Path	Specifies the path to the RScript that will run when the policy is executed.	You can type the path manually, or you can browse to it by clicking the <input type="button" value="..."/> button.
Additional Property fields	Specify an input for the RScript that will be used to calculate a result.	The number of Property fields displayed in the Properties window is determined by the number of properties specified in the selected RScript.

Example of the R-Script Node

The following example illustrates how you can use an R-Script node to evaluate certain values in other nodes and return a particular value. Consider the following nodes and connection.



As shown in the following image, you can use the **Properties** window for the R-Script node to select a particular RScript that was created using the RScript Builder. This RScript accepts input for the *Liner Base DF* and *Lining Condition Adjustment* variables, which are defined in the RScript itself. In this example, the *Liner Base DF* property and the *Lining Condition Adjustment* property (which correspond to the variables) in the **Properties** window are set to receive values from the *Inorganic Base DF Lookup* Entity node and the *Lining Condition Adjustment Look up* Entity node, respectively. In the case of the Inorganic Base DF Lookup Entity node, the value is from the Damage Factor field of the Inorganic Base DF Lookup family. In the case of the Lining Condition Adjustment Lookup Entity node, the value is from the Adjustment Multiplier field of the Lining Condition Adjustment Lookup family.

The RScript selected in the R-Script node uses the input values to calculate and return a value. In this example, the Return Value node will return the value calculated by the R-Script node for the policy.

R-Script Nodes

Properties	
ID	n14
Name	Damage Factor Calculation
RScript Path	<input type="checkbox"/> Field Public\Meridium\Modules\Risk Ba ...
LinerBaseDF	<input checked="" type="checkbox"/> Field Inorganic Base DF Lookup ▼ Damage Factor ▼
LiningConditionAdjustment	<input checked="" type="checkbox"/> Field Lining Condition Adjustment Look ▼ Adjustment Multiplier ▼

Collection Filter Nodes

A *Collection Filter* node is a [Calculation node](#) which you can use to apply one or more filters to collections that are represented by nodes in the policy diagram. Filters may, for example, specify a range of values or [dates](#) within which readings must fall in order to be evaluated by successor nodes. The following image shows what the Collection Filter node looks like.





The input of a Collection filter node must be a collection of data. The output of a Collection Filter node, *Filtered Collection*, includes only the rows in a collection that pass *all* filter criteria that is defined in the Collection Filter's **Properties** window.

Properties of the Collection Filter node

The following image shows what the **Properties** window looks like for a Collection Filter node after you add the node to the policy diagram.

The **Properties** window for a Collection Filter node contains the items that are described in the following table.

Item	Description	Notes
Collection section	Specifies the collection that you want to filter.	You will use the Field check box and subsequent options to define the collection in this section.

Item	Description	Notes
Add Filter button	Add a new filter row to the Collection Filter node.	Each filter row consists of a column, operator, condition, and  button.
Column list	Specifies the column whose values you want to use in the filter.	This list contains the columns that are available in the selected collection.
Operator list	Specifies the comparison operator that you want to apply to the values in the selected column.	This list contains the following operators: <ul style="list-style-type: none"> • Greater than (>) • Greater than or equal (>=) • Less than (<) • Less than or equal (<=) • Equal (=) • Does not equal (!=)
Condition value	The value that will be compared against the values in the corresponding column to determine the output of the node.	You will use the Field check box and subsequent options to define the value in this section.
 button	Deletes the corresponding filter row from the Collection Filter node.	None

Example of the Collection Filter Node

The following example illustrates how you can use the Collection Filter node to filter query results such that only certain results are used in subsequent calculations. Consider the following nodes and connection.



In this example, a [Query node](#), *Policy Events Query*, is connected to a Collection Filter node.

As shown in the following image, you can use the **Properties** window of the Collection Filter node to define multiple filters that are applied to the full table of results from the query. Specifically, the query results are filtered to include only the policy events that

lasted longer than ten seconds, occurred within the last month, and have the name *IOW Excursion*.

The screenshot shows the 'Properties' window for a 'Collection Filter' node. The 'ID' is 'n11'. The 'Name' is 'Collection Filter'. Under the 'Collection' section, 'Field' is checked. The 'Policy Events Query' is set to 'Policy Events Query' and the 'Result Set' is 'Result Set'. There is a '+ Filter' button. Below this is a table with three filter conditions:

Column	Operator	Condition
Duration (seconds)	>	Field 10
Start Time	>	Field now-1 month
Name	=	Field IOW Excursion

The output of the Collection Filter node, i.e. the filtered results of the query, can then be used by successor nodes to perform a variety of calculations.

[Click here](#) for an example of a Collection Filter node used within a complete policy diagram.

Last Nodes

A *Lastnode* is a [Calculation node](#) which you can use in a policy diagram to retrieve the last row in a collection, whose fields you can then use in successor nodes to perform various calculations or actions. The following image shows what a Last node looks like.



The input for a Last node must be a collection. The output for the Last node is any field in the row that the Last node retrieves. For reading collections associated with Measurement Location, Health Indicator, or Xi Tag nodes, the last row is the *most recent* reading. For collections associated with Query nodes, the last row depends on the order in which the query is sorted.

Properties of the Last Node

The following image shows what the **Properties** window looks like for a Last node after you add the node to the policy diagram.

 A screenshot of a "Properties" window. The window has a title bar with "Properties" and a close button. It contains several fields:

- ID: n9
- Name: Last 9
- Collection: A section with a checkbox labeled "Field" and an empty text input field below it.

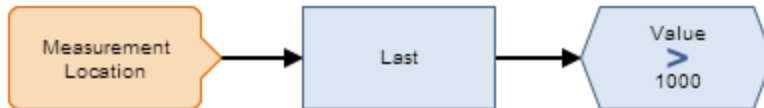
The **Properties** window for a Last node contains the items that are described in the following table.

Item	Description	Notes
Collection section	Specifies the collection from which you want to retrieve the last row.	You will use the Field check box and subsequent options to define the value in this section.

Example of the Last Node

The following example illustrates how you can use a Last node to retrieve the most recent reading associated with a Measurement Location record and use this value in a subsequent calculation. Consider the following nodes and connections.

Last Nodes



As shown in the following image, you can use the **Properties** window for the Last node to select the Measurement Location node and the *Readings* option. When this option is selected, the Last node evaluates the readings that are associated with the Measurement Location record and returns the last row in the collection, which contains the most recent reading.

Properties	
ID	n16
Name	
Last	
Collection	
<input checked="" type="checkbox"/> Field	
Measurement Location	▼
Readings	▼

You can use the most recent reading in various subsequent calculations. In this example, the Greater Than node evaluates the value of the most recent reading to determine whether or not it is greater than 1,000.

[Click here](#) for an example of a Last node used within a complete policy diagram.

Min and Max Nodes

Min and *Max* nodes are [Calculation nodes](#) which you can use in a policy diagram to determine the row of a collection which contains the smallest or largest value, respectively, in a specified column. The following image shows what a Max node looks like.



The input for a Min or Max node must be a collection with a column containing numeric or time-based values. The output of a Min or Max node is any field in the row that the Min or Max node retrieves. If there is more than one point with the same minimum or maximum value in the range, then the row with the most recent timestamp is returned.

Properties of Min and Max Nodes

The following image shows what the **Properties** window looks like for a Max node after you add the node to the policy diagram.

 A screenshot of a "Properties" window for a Max node. The window has a title bar with "Properties" and a close button. It contains several fields:

- ID:** n10
- Name:** Max 2
- Collection:** A section with a checked checkbox labeled "Field" and an empty text input field below it.
- Collection Column:** A dropdown menu with "select..." selected.

The **Properties** windows for Min and Max nodes contain the items that are described in the following table.

Item	Description	Notes
Collection section	Specifies the collection for which you want to determine the row containing the smallest or largest value in the column selected in the Collection Column section.	You will use the Field check box and subsequent options to define the value in this section.

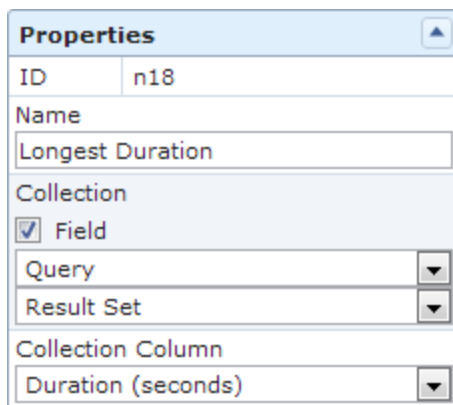
Item	Description	Notes
Collection Column section	Specifies the column that contains the values from which you want to determine the smallest or largest value.	This list contains the columns that are available in the selected collection. The column that you select must contain numeric or time-based values.

Example of the Max Node

The following example illustrates how you can use a Max node to determine the row of the query has the largest value in the Duration field, and then use a successor node to evaluate a different field in the same row. Consider the following nodes and connections.



As shown in the following image, you can use the **Properties** window for the *Longest Duration* Max node to select the Query node and the *Result Set* option. The *Duration (seconds)* column of the Result Set is also selected, indicating that the query row with the largest value in the Duration (seconds) column will be returned.



As shown in the following image, you can then use the **Properties** window for the Equal node to determine if the Severity field in the query row with the longest duration contains the value *Warning*.

Min and Max Nodes

Properties	
ID	n23
<input checked="" type="checkbox"/> Field	
Longest Duration	▼
Severity	▼
Display:	Field ▼
=	
<input type="checkbox"/> Field	
Warning	

Average Nodes

An *Average node* is a [Calculation node](#) which you can use in a policy diagram to calculate the average value of data in a specified column of a collection. The following image shows what an Average node looks like.



The input for an Average node must be a collection with a column containing numeric or time-based values. The output of an Average node, *Value*, contains the average value of data in the specified column.

Properties of the Average Node

The following image shows what the **Properties** window looks like for an Average node after you add the node to the policy diagram.

 A screenshot of a "Properties" window. The window has a title bar with "Properties" and a close button. Below the title bar, there are several fields:

- ID:** n23
- Name:** Average 1
- Collection:** A section with a checkbox labeled "Field" (which is unchecked) and an empty text input field below it.
- Collection Column:** A dropdown menu showing "select..."

The **Properties** window for an Average node contains the items that are described in the following table.

Item	Description	Notes
Collection section	Specifies the collection that contains the values that you want to average.	You will use the Field check box and subsequent options to define the value in this section.
Collection Column section	Specifies the column that contains that values that you want to average.	This list contains the columns that are available in the selected collection. The column that you select must contain numeric or time-based values.

Example of the Average Node

Average Nodes

The following example illustrates how you can use an Average node to evaluate the readings that are associated with a Health Indicator record and return the average reading value. Consider the following nodes and connection.



As shown in the following image, you can use the **Properties** window for the Average node to select the Health Indicator node and the *Readings* option. The *Value* column of the Readings collection is also selected, indicating that the output of the node is the average of the values in the in the Value column.

Properties	
ID	n24
Name	
Average Reading Value	
Collection	
<input checked="" type="checkbox"/>	Field
Health Indicator	
Readings	
Collection Column	
Value	

Sum Nodes

A *Sum node* is a [Calculation node](#) which you can use in a policy diagram to calculate the total value of data in a specified column of a collection. The following image shows what the Sum node looks like.



The input for a Sum node must be a collection containing numeric or time-based values. The output of a Sum node, *Value*, contains the result of the calculation for the specified column.

Properties of the Sum Node

The following image shows what the **Properties** window looks like for a Sum node after you add the node to the policy diagram.

 A screenshot of a "Properties" window. The window has a title bar with the word "Properties" and a close button. Below the title bar, there are several fields:

- ID:** n0
- Name:** Sum 0
- Collection:** A section with a checkbox labeled "Field" (which is unchecked) and an empty text input field below it.
- Collection Column:** A dropdown menu with "select..." selected.

The **Properties** window for a Sum node contains the items that are described in the following table.

Item	Description	Notes
Collection section	Specifies the collection that contains the values for which you want to calculate the total value.	You will use the Field check box and subsequent options to define the value in this section.
Collection Column section	Specifies the column that contains that values for which you want to calculate the total value.	This list contains the columns that are available in the selected collection. The column that you select must contain numeric or time-based values.

Example of the Sum Node

Sum Nodes

The following example illustrates how you can use a Sum node in the policy diagram to determine the total value of the Measurement Location readings that were taken within the last week. Consider the following nodes and connections.



In this example, the Collection Filter node is used to filter the readings associated with the Measurement Location node to only the readings that occurred within the last week. The Sum node is then used to calculate the total of the values in the Value column of the filtered collection. The following image shows the what **Properties** window looks like for the Sum node.

Properties	
ID	n15
Name	Sum
Collection	
<input checked="" type="checkbox"/> Field	
Within Last Week	▼
Filtered Collection	▼
Collection Column	
Value	▼

The remaining nodes indicate that a recommendation should be created if the total value of the readings is less than or equal to 100.

Count Nodes

A *Count node* is a [Calculation node](#) which you can use in a policy diagram to calculate the total number of rows in a collection. The following image shows what the Count node looks like.



The input for a Count node must be a collection. The output for a Count node, *Value*, contains the result of the calculation.

Properties of the Count Node

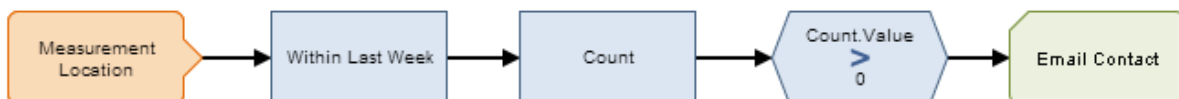
The following image shows what the **Properties** window looks like for a Count node after you add the node to the policy diagram.

The **Properties** window for a Count node contains the items that are described in the following table.

Item	Description	Notes
Collection section	Specifies the collection for which you want to calculate the total number of rows.	You will use the Field check box and subsequent options to define the value in this section.

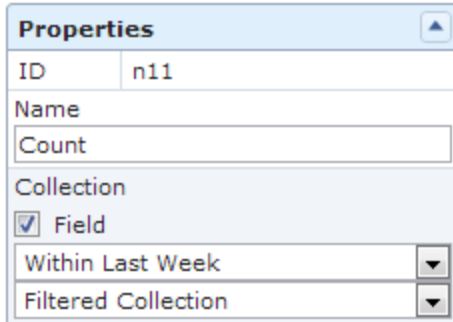
Example of the Count Node

The following example illustrates how you can use a Count node in the policy diagram to determine if any Measurement Location readings were recorded within the last week. Consider the following nodes and connections.



Count Nodes

In this example, the Collection Filter node is used to filter the readings associated with the Measurement Location node to only the readings that occurred within the last week. The Count node is then used to calculate the number of readings that occurred. The following image shows what the **Properties** window for the Count node looks like.



The image shows a 'Properties' window for a Count node. It has a title bar with the word 'Properties' and a small upward-pointing arrow. The window contains several fields and controls:

- ID:** n11
- Name:** Count
- Collection:** A section containing:
 - Field
 - Within Last Week (with a dropdown arrow)
 - Filtered Collection (with a dropdown arrow)

The remaining nodes indicate that an email message should be sent if the number of readings is greater than zero.

Threshold Statistics Nodes

A *Threshold Statistics node* is a [Calculation node](#) that you can use in a policy diagram to determine the frequency and duration over which input values cross a defined threshold (i.e. meet a defined condition). The following image shows what a Threshold Statistics node looks like.



The input for a Threshold Statistics node must be a collection with columns containing timestamps and numeric values. A Threshold Statistics node generates the following outputs:

- *Count*, which represents the number of times that input values crossed the defined threshold.
- *Accumulated Time*, which represents the [amount of time](#) over which input values crossed the defined threshold.

Properties of the Threshold Statistics Node

The following image shows what the **Properties** window looks like for a Threshold Statistics node after you add the node to the policy diagram.

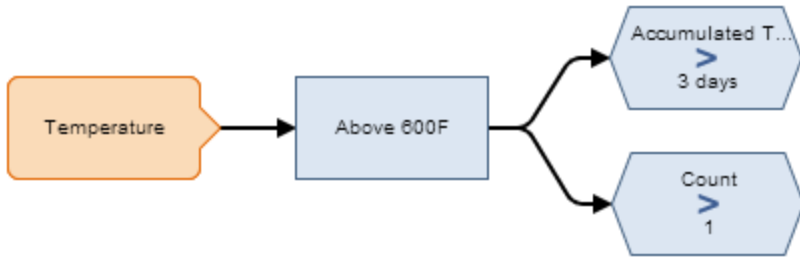
Properties	
ID	n12
Name	Threshold Statistics 7
Collection	<input type="checkbox"/> Field
Timestamp Column	select...
Value Column	select...
Threshold	<input type="checkbox"/> Field
Operator	select...

The **Properties** window for a Threshold Statistics node contains the items that are described in the following table.

Item	Description	Notes
Collection section	Specifies the collection for which you want to calculate a threshold statistic.	You will use the Field check box and subsequent options to define the value in this section.
Timestamp Column list	Specifies the column that contains the timestamps that you want to use to calculate the accumulated time during which the input values cross the defined threshold.	This list contains the columns that are available in the selected collection.
Value Column list	Specifies the column that contains the input values that will be compared to the threshold value.	This list contains the columns that are available in the selected collection.
Threshold section	Specifies the threshold value that will be compared to the values that are defined in the Value Column list.	You will use the Field check box and subsequent options to define the value in this section.
Operator section	Specifies the operator that will be used to compare inputs values to the threshold value.	<p>This list contains the following operators:</p> <ul style="list-style-type: none"> • Less than (<) • Less than or equal (<=) • Equal (=) • Greater than or equal (>=) • Greater than (>) <p>The Equal operator is the only operator that can be used with character values. For all other operators, the values that you select in the Value Column list and the Threshold section must be numeric.</p>

Example of the Threshold Statistics Node

The following example illustrates how you can use a Threshold Statistics node with Condition nodes to evaluate how many times and for how long the reading values that were recorded in the process historian exceeded a defined threshold. Consider the following nodes and connections.



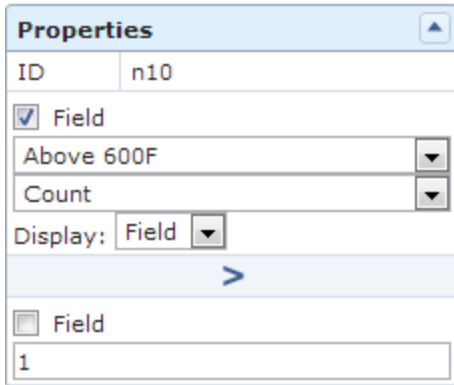
In this diagram, the *Temperature* Xi Tag node is connected to the *Above 600 F* Threshold Statistics node, which is also connected to two Greater Than nodes: *Accumulated Time > 3 Days* and *Count > 1*.

As shown in the following image, you can use the **Properties** window for the Threshold Statistics node to:

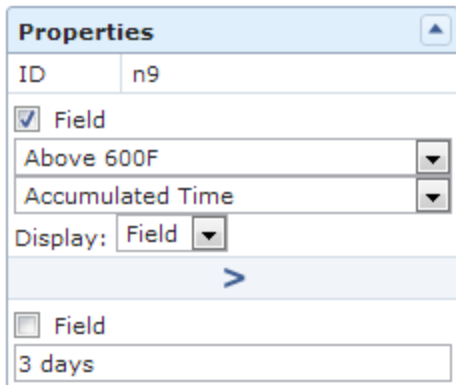
- Indicate that you want to evaluate the collection of *HDA Readings* associated with an Xi Tag record against some threshold.
- Identify that the *Time stamp* and *Value* columns of the *HDA Readings* collection store the timestamps and values that you want to evaluate.
- Indicate that the values in the *Value* column will be evaluated to determine whether or not they are greater than or equal to 600.

Properties	
ID	n8
Name	Above 600F
Collection	
<input checked="" type="checkbox"/>	Field
	Temperature
	HDA Readings
Timestamp Column	
	Timestamp
Value Column	
	Value
Threshold	
<input type="checkbox"/>	Field
	600
Operator	
	>=

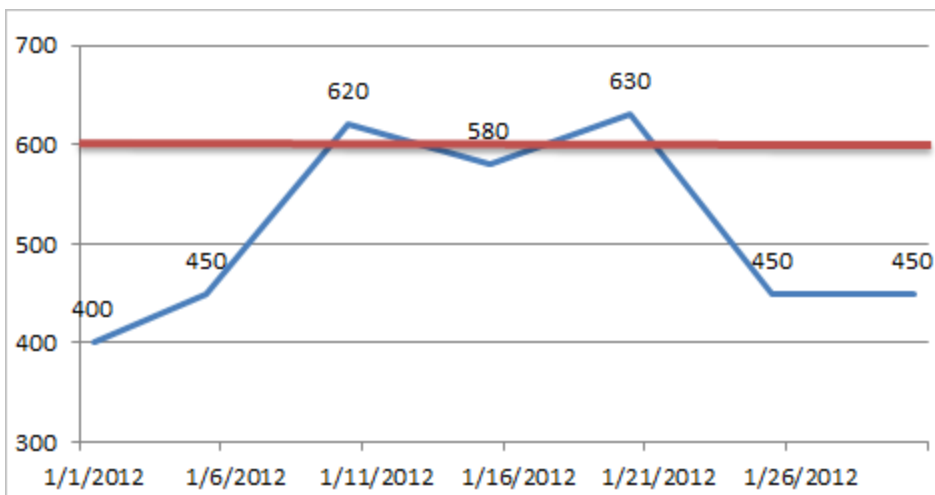
As shown in the following image, you can use the **Properties** window for the *Count > 1* Greater Than node to select the input *Count* from the Threshold Statistics node.



As shown in the following image, you can use the **Properties** window for the *Accumulated Time > 3 Days Greater Than* node to select the input *Accumulated Time* from the Threshold Statistics node.

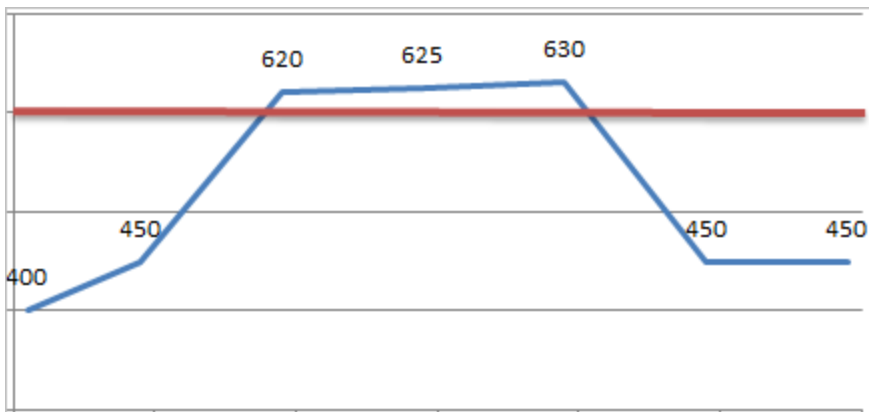


When these options are selected, the condition nodes evaluate the number of times and amount of time over which the reading values recorded in the process historian exceeded the threshold of 600 degrees Fahrenheit. To illustrate this example, consider the following graph, where the x-axis displays dates, the y-axis displays reading values, and the blue line displays reading values that were recorded by the process historian on specific dates. Note also that the red horizontal line identifies the threshold value 600.



In the graph, you can see that the process historian recorded a reading value above 600 *two* times, where one value was 620 (on 1/10) and the other value was 630 (on 1/20). These values are treated as two separate occurrences of exceeding the threshold value because the reading value actually dropped below 600 between those two dates. In this scenario, the condition *Count > 1* would be true, but the condition *Accumulated Time > 3 Days* would be false.

Now consider the following graph in which the value recorded on 1/15/2012 was also above 600. In this scenario, the Meridium APM system determines that the threshold value was exceeded only *one* time, not three times. Specifically, whenever a reading value exceeds the threshold value, if the threshold value *continues* to be exceeded with each subsequent reading value, the Meridium APM system treats it as *one* occurrence of exceeding the threshold. Whenever a subsequent reading value drops below the threshold value, the next time another reading value exceeds it again, the Meridium APM system treats that as a separate occurrence.



Here, you can see that the first reading value that exceeded the threshold value is 620. After that, two consecutive reading values also exceeded the threshold: 625 and 630. These additional reading values, however, are not treated as a second occurrence of exceeding the threshold value because they are consecutive. Therefore, the condition *Count > 1* would be false and the condition *Accumulated Time > 3 Days* would be true.

[Click here](#) for an example of a Threshold Statistics node used within a complete policy diagram.

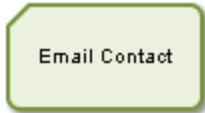
About Action Nodes

Action nodes represent actions that will be performed when a policy is executed and the conditions preceding the Action node are met. You can add several types of Action nodes to the policy diagram using the [Actions toolbar](#) and provide specific information for each node using the [Properties window](#). Most Action nodes are shaded green in the policy diagram. The following Action nodes are available:

- [Email Contact](#)
- [Create Recommendation](#)
- [Create Event](#)
- [Close Event](#)
- [Rule](#)
- [Return Value](#)
- [Deactivate This Instance](#)
- [Add Value to Health Indicator](#)
- [Create Production Event](#)

Email Contact Nodes

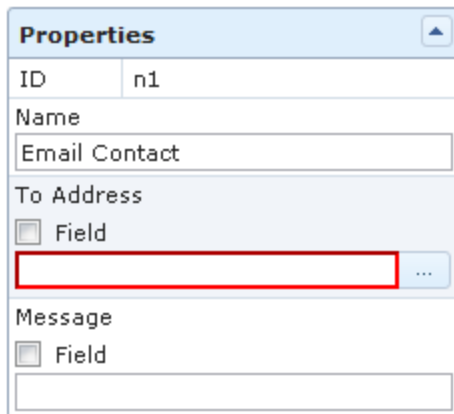
An *Email Contact node* represents an action to send an email message. The following image shows what an Email Contact node looks like.




When an Email Contact node is executed, an email message will be sent to the recipient (s) that are specified on the **Properties** window for the Email Contact node.

Properties of the Email Contact Node

The following image shows what the **Properties** window looks like for an Email Contact node after you add the node to the policy diagram.

A screenshot of a "Properties" dialog box for an Email Contact node. The dialog has a title bar with "Properties" and a close button. It contains several sections: "ID" with a text field containing "n1"; "Name" with a text field containing "Email Contact"; "To Address" section with a checkbox labeled "Field" (unchecked) and a text field containing an empty address, with a red rectangle highlighting the text field and a "..." button to its right; and "Message" section with a checkbox labeled "Field" (unchecked) and a text field containing an empty message body.

The **Properties** window for an Email Contact node contains the items that are described in the following table.

Item	Description	Notes
To Address section	The email address(es) to which the message should be sent.	<p>You will use the Field check box and subsequent options to define the value in this section.</p> <p>When the Field check box is <i>cleared</i>, you can type one or more email addresses directly in the text box, or you can click the  button to display the Choose User window.</p> <p>If you specify more than one email address in the text box, each email address should be separated by a comma or semicolon.</p>
Message section	The content of the email message.	You will use the Field check box and subsequent options to define the value in this section.

Using the Choose User Window

You can use the **Choose User** window to select one or more Human Resource records that are associated with the people to whom you want to send the message. The following image shows what the **Choose User** window looks like.


Last Name	First Name	Email Address
Demo	Joe	mdaniel@meridium.com
Leader	Package	dbrown@meridium.com

The **Choose User** window contains a grid that displays the results of the [Email Address Query](#). If you want to filter the results that appear in the grid, you can use the text boxes and the **Find Now** button above the grid. The following columns are available in the grid:

- **Last Name**
- **First Name**
- **Email Address**

The following items are displayed below the grid:

- Information and options about the results. These options mimic the options that are displayed at the bottom of the results section of the Search Tool.
- The following buttons:
 - **Choose:** Populates the text box in the **To Address** section of the Email Contact node with the value in the Email Address field for the selected Human Resource record(s). This button is enabled only when one or more Human Resource records are selected in the grid. When you select more than one Human Resource record and click the **Choose** button, each Email Address value populated in the text box in the **To Address** section will be separated by a semicolon.
 - **Cancel:** Closes the **Choose User** window.

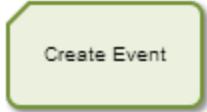
You can also close the window by clicking the  button in the top, right corner of the window.

Example of the Email Contact Node

[Click here](#) for an example of an Email Contact node used within a complete policy diagram.

Create Event Nodes

A *Create Event node* represents an action to create a [Policy Event record](#), which stores information about events that are associated with Equipment or Functional Location records. The following image shows what a Create Event node looks like.



When a Create Event node is executed, a [Policy Event record](#) will be created using the values that you specify on the **Properties** window for the node. One Policy Event record will be created for each Create Event node in the policy diagram with a unique name (i.e., the value in the **Event Name** section on the **Properties** window).

If a Policy Event record is created for an event that has a duration, additional Policy Event records with the same name will *not* be created until the event is closed (i.e. the **End Time** field in the Policy Event record is populated with the time and date associated with the end of the event).

Properties of the Create Event Node

The following image shows an example of what the **Properties** window looks like for a Create Event node after you add the node to the policy diagram.

Create Event Nodes

Properties	
ID	n3
Name	Create Event
Asset	Select Node...
Event Name	<input type="checkbox"/> Field
Description	<input type="checkbox"/> Field
Event Type	Generic
Severity	Information
Start Time	<input type="checkbox"/> Field
End Time	<input type="checkbox"/> Field
Close Description	<input type="checkbox"/> Field
Has Duration	<input type="checkbox"/> Field <input type="radio"/> Yes <input checked="" type="radio"/> No
Time Line Reset	<input type="checkbox"/> Field <input type="radio"/> Yes <input checked="" type="radio"/> No

The **Properties** window for a Create Event node contains the items that are described in the following table. The values that you define in each section will be used to populate the corresponding fields in the [Policy Event record](#) that is created.

Item	Description	Notes
Asset list	Specifies the node representing the Equipment or Functional Location record that is associated with the event, (i.e., the Equipment or Functional Location record to which the Policy Event record will be linked).	This value is required. This list contains the names of the nodes that are predecessors to the Create Event node.
Event Name section	Specifies the name of the event.	This value is required. You will use the Field check box and subsequent options to define the value in this section. The value that you specify is displayed in the policy diagram in parentheses below the node name.
Description section	Specifies a description of the event.	You will use the Field check box and subsequent options to define the value in this section.
Event Type list	Specifies the type of event. You can select one of the following values: <ul style="list-style-type: none"> • Generic: Indicates that the event is not associated with a specific type of event. • Excursion: Indicates that the event is associated with an operation that is outside of an established operating window. 	The value <i>Generic</i> is selected by default.

Item	Description	Notes
Severity list	<p>Specifies the severity of the event. You can select one of the following values:</p> <ul style="list-style-type: none">• Information: Indicates that the event exists simply to provide information about activity conducted for the associated piece of equipment or location. For example, if the event is triggered by conditions that call for a routine inspection on a piece of equipment, you might select <i>Information</i> in this list.• Warning: Indicates that the event exists to communicate some sort of low-risk warning about the associated piece of equipment or location. For example, if the event is triggered by conditions that indicate that a piece of equipment is nearing its end of life, you might select <i>Warning</i> in this list.• Alert: Indicates that the event exists to communicate some sort of high-risk warning about the associated asset. For example, if the event is triggered by conditions that indicate that a piece of equipment is about to fail, you might select <i>Alert</i> in this list.	The value <i>Information</i> is selected by default.

Item	Description	Notes
Start Time section	Specifies the timestamp that is associated with the beginning of the event.	<p>This value is required.</p> <p>You will use the Field check box and subsequent options to define the value in this section. If you type a date, you must use the correct format.</p>
End Time section	Specifies the timestamp that is associated with the end of the event.	<p>You should only define a value in this section when the Has Duration field is set to <i>True</i>.</p> <div data-bbox="948 646 1398 940" style="border: 1px solid red; padding: 5px;"> <p>IMPORTANT: If the policy contains a Close Event node, you should <i>not</i> specify values in this section. You should specify values in the End Time section on the Properties window that is displayed for the Close Event node.</p> </div> <p>You will use the Field check box and subsequent options to define the value in this section. If you type a date, you must use the correct format.</p>
Close Description section	Specifies a description of why the event was closed.	<div data-bbox="948 1157 1398 1451" style="border: 1px solid red; padding: 5px;"> <p>IMPORTANT: If the policy contains a Close Event node, you should <i>not</i> specify values in this section. You should specify values in the Close Description section on the Properties window that is displayed for the Close Event node.</p> </div> <p>You will use the Field check box and subsequent options to define the value in this section.</p>

Item	Description	Notes
Has Duration section	Specifies whether or not there is an end time associated with the event.	You will use the Field check box and subsequent options to define the value in this section. The value that you define must be a logical result (i.e., <i>Yes</i> or <i>No</i>). <div style="border: 1px solid red; padding: 5px;"> IMPORTANT: If the policy contains a Close Event node, you must set the value in this field to <i>True</i>. </div>
Time Line Reset section	This field is not currently used.	None

Example of the Create Event Node

[Click here](#) for an example of a Create Event node used within a complete policy diagram.

Close Event Nodes

A *Close Event node* represents an action to populate fields in an existing [Policy Event record](#) to signify that the associated event has been closed. You can add a Close Event node to a policy that already contains a [Create Event node](#) (which will create a Policy Event record when policy conditions are met). The following image shows what a Close Event node looks like.



When a Close Event node is executed, the following fields will be populated in the *associated* [Policy Event record](#):

- End Time
- Close Description

The *associated* [Policy Event record](#) is the record that was created as a result of the execution of a policy containing a Create Event node whose:

- Name value matches the value in the **Event Name** section on the **Properties** window for the Close Event node.
- Has Duration field is set to *True*.
- Start Time field contains a value.
- End Time field does *not* contain a value.

In the policy diagram, a dotted line will appear between Create Event and Close Event nodes that are associated with the same [Policy Event record](#) (i.e., with the same value specified in the **Event Name** section on the **Properties** window for each node).

Properties of the Close Event Node

The following image shows an example of what the **Properties** window looks like for a Close Event node after you add the node to the policy diagram.

The **Properties** window for a Close Event node contains the items that are described in the following table. You will use the [Field check box and subsequent options](#) to define the values in each section. The values that you define in each section will be used to populate the corresponding fields in the associated [Policy Event record](#).

Item	Description	Notes
Event Name section	Specifies the name of the event that you want to close. <div style="border: 1px solid red; padding: 5px; margin: 5px 0;"> IMPORTANT: The value(s) that you specify in this section must exactly match the value(s) specified in the Event Name section on the Properties window that is displayed for the associated Create Event node (i.e. the value in the Name field in the associated Policy Event record). </div>	The value that you specify is displayed in the policy diagram in parentheses below the node name.
End Time section	Specifies the timestamp that is associated with the end of the event.	This value is required. If you type a date, you must use the correct format .

Item	Description	Notes
Close Description section	Specifies a description of why the event was closed.	None

Example of the Close Event Node

[Click here](#) for an example of a Close Event node used within a complete policy diagram.

Create Recommendation Nodes

A *Create Recommendation node* represents an action to create a [Policy Recommendation record](#). The following image shows what a Create Recommendation node looks like.



When a Create Recommendation node is executed, a [Policy Recommendation record](#) will be created using the values that you specify on the **Properties** window for the node.

Properties of the Create Recommendation Node

The following image shows an example of what the **Properties** window looks like for a Create Recommendation node after you add the node to the policy diagram.

Create Recommendation Nodes

Properties	
ID	n0
Name	Create Recommendation
State Assignee User ID	<input type="checkbox"/> Field
Associated Reference	<input type="checkbox"/> Field
Create Work Request?	<input type="checkbox"/> Field <input type="radio"/> Yes <input checked="" type="radio"/> No
Equipment ID	<input type="checkbox"/> Field
Equipment Technical Number	<input type="checkbox"/> Field
Functional Location ID	<input type="checkbox"/> Field
Recommendation Headline	<input type="checkbox"/> Field
Recommendation Priority	<input type="checkbox"/> Field
Target Completion Date	<input type="checkbox"/> Field

The **Properties** window for a Create Recommendation node contains the items that are described in the following table. You will use the [Field check box and subsequent options](#) to define the values in each section. The values that you define in each section will be used to populate the corresponding fields in the [Policy Recommendation record](#) that is created.

Item	Description	Notes
State Assignee User ID section	Specifies the user that is assigned to the initial state.	The state assignee value must match <i>exactly</i> a valid User ID for an active Security User.
Associated Reference section	Specifies the Reference ID of the event or entity that originated the Recommendation.	This section corresponds to the Originating Reference field in a Policy Recommendation record.
Create Work Request section	Specifies whether or not a work request for the EAM system that you have configured in Meridium APM will be created from the Policy Recommendation record.	The value in this section must be a logical result (i.e., <i>Yes</i> or <i>No</i>).
Equipment ID section	The Record ID of the Equipment record to which the Policy Recommendation record is linked.	None
Event Start Date section	Specifies the time stamp that is associated with the beginning of the event for which the Policy Recommendation record is created.	If you type a date, you must use the correct format .
Functional Location ID section	<p>The Record ID of the Functional Location record to which the Policy Recommendation record is linked.</p> <p>-or-</p> <p>The Record ID of the Functional Location record that is linked to the Equipment record to which the Policy Recommendation record is linked.</p>	None

Item	Description	Notes
Recommendation Headline section	A short description of the recommended action.	None
Recommendation Priority section	Specifies a priority value used to rank importance of the recommendation.	If you type a value, it must be a valid value according to any field-level rules that you have specified for the Recommendation Priority field.
Target Completion Date section	The date by which the recommended action should be completed.	This value is required. If you type a date, you must use the correct format .

Note: If you add a custom field to the Policy Recommendation family, that field will also appear in the **Properties** window for the Create Recommendation node.

Example of the Create Recommendation Node

[Click here](#) for an example of a Create Recommendation node used within a complete policy diagram.

Rule Nodes

A *Rule node* represents an action to execute a custom rule that provides functionality that the existing nodes in Policy Manager do not provide. The following image shows what a Rule node looks like.



IMPORTANT: The improper implementation of a rule through the Rule node could severely impact the performance of policy executions. If you want to use this node, you should contact Meridium, Inc. for additional instructions and assistance.

Properties of the Rule Node

The following image shows what the **Properties** window looks like for a Rule node after you add the node to the policy diagram.

The **Properties** window for a Rule node contains the items that are described in the following table.

Item	Description	Notes
Rule Path text box	Specifies the Catalog path to the rule project that contains the rule that should be executed when the policy is executed.	You must type the path manually (e.g., type <code>Public\Rules_Library\Client\Default Rules\AQA_IEU_Rules</code>).
Rule Class list	Specifies the class containing the rule that should be executed when the policy is executed.	

Return Value Nodes

A *Return Value node* represents an action to return a specific value. You can use this node to include specified values in the [execution results](#) of a policy. The following image shows what a Return Value node looks like.



When a policy is executed, if more than one Return Value node with the same name is executed, only one of these nodes will return a value. Therefore, when using Return Value nodes in a policy diagram, you should:

- Specify a unique name for each Return Value node.
- or-
- Define policy logic such that only one Return Value node with a given name can be executed when the policy is executed.

Properties of the Return Value Node

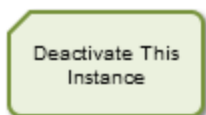
The following image shows an example of what the **Properties** window looks like for a Return Value node after you add the node to the policy diagram.

The **Properties** window for a Return Value node contains the items that are described in the following table.

Item	Description	Notes
Return Value section	Specifies the value that you want to return.	You will use the Field check box and subsequent options to define the value in this section.

Deactivate This Instance Nodes

A *Deactivate This Instance node* represents an action to deactivate the current policy instance after an action is taken. You can use this node to ensure that an action is taken only once per instance. The following image shows what a Deactivate This Instance node looks like.

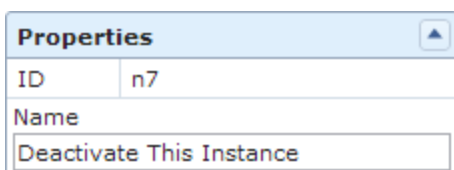


When a Deactivate This Instance node is executed, the policy instance whose values caused the node to be executed will be deactivated automatically. If more than one policy instance is associated with the policy, all other policy instances will remain unchanged.

Once the conditions that caused the instance to be deactivated have been addressed, you can reactivate the policy instance manually from the [Instances tab in Policy Manager](#) or the Policies tab in Asset Health Manager.

Properties of the Deactivate This Instance Node

The following image shows what the **Properties** window looks like for a Deactivate This Instance node after you add the node to the policy diagram.



Other than specifying a name for the node, there are no properties to configure for a Deactivate This Instance Node.

Example of the Deactivate This Instance Node

The following example illustrates how you can use a Deactivate This Instance node to ensure that only one Policy Recommendation record is created as a result of the policy conditions being met for a specific policy instance. Consider the following nodes and connection.



In this example, the Xi Tag node and the Condition node to which it is connected indicate that a Policy Recommendation record should be created when the DA Reading Value associated with the Xi Tag node exceeds 200.

Consider that the following policy instances are associated with the Xi Tag node:

Deactivate This Instance Nodes

- Pump 101, whose DA Reading Value *is* greater than 200.
- Pump 102, whose DA Reading Value *is not* greater than 200.

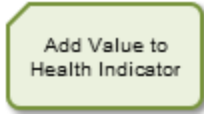
Based on this policy diagram, when the policy is executed, a Policy Recommendation record for Pump 101 will be created. Because the policy includes a *Deactivate This Instance* node that is connected as a successor to the Create Recommendation node, after the Policy Recommendation record is created, the Meridium APM system will deactivate *automatically* the policy instance that is associated with Pump 101. The policy instance that is associated with Pump 102 will remain active.

Because the policy instance that is associated with Pump 101 is no longer active, even if the DA Reading Value associated with the pump continues to exceed 200, additional Policy Recommendation records will *not* be created.

[Click here](#) for an example of a Deactivate This Instance node used within a complete policy diagram.

Add Value to Health Indicator Nodes

An *Add Value to Health Indicator* node represents an action to create a Health Indicator Value record and link it to a [Health Indicator record created via Policy Manager](#). These records can be viewed and managed in the Asset Health Manager (AHM) module. The following image shows what an Add Value to Health Indicator node looks like.



When an Add Value to Health Indicator node is executed, a Health Indicator Value record will be created using the timestamp and numeric value that you specify on the **Properties** window for the node. The new Health Indicator Value record will be linked to a Health Indicator record that you specify.

Note: If the Health Indicator record to which you want to add values does not exist, you can [create a new Health Indicator record via the Instances Tab](#) of the Health Indicator node to which the Add Value to Health Indicator node is connected.

Properties of the Add Value to Health Indicator Node

The following image shows what the **Properties** window looks like for an Add Value to Health Indicator node after you add the node to the policy diagram.

A screenshot of a "Properties" window for an "Add Value to Health Indicator" node. The window has a title bar with "Properties" and a close button. It contains several sections: "ID" with the value "n9"; "Name" with the text "Add Value to Health Indicator"; "Health Indicator Record" with a dropdown menu showing "Select Node..."; "Timestamp" with a checkbox labeled "Field" and an empty text box; and "Value (Numeric)" with a checkbox labeled "Field" and an empty text box.

The **Properties** window for an Add Value to Health Indicator node contains the items that are described in the following table.

Item	Description	Notes
Health Indicator Record list	Specifies the node that represents the Health Indicator record to which the new Health Indicator Value record will be linked.	The node that you select must represent a Health Indicator record that was created via Policy Manager .
Timestamp section	Specifies the timestamp of the new Health Indicator Value record. This value is stored in the Timestamp field in the Health Indicator Value record.	You will use the Field check box and subsequent options to define the value in this section. The value in this section must be a timestamp.
Value (Numeric) section	Specifies the value of the new Health Indicator Value record. This value is stored in the Value (Numeric) field in the Health Indicator Value record.	You will use the Field check box and subsequent options to define the value in this section. The value in this section must be numeric.

Example of the Add Value to Health Indicator Node

[Click here](#) for an example of an Add Value to Health Indicator node used within a complete policy diagram.

Create Production Event Nodes

A *Create Production Event node* represents an action to create a Production Event record. The following image shows what a Create Production Event node looks like.



When a Create Production Event node is executed, a Production Event record will be created using the values that you specify on the **Properties** window for the node.

Properties of the Create Production Event Node

The following image shows an example of what the **Properties** window looks like for a Create Production Event node after you add the node to the policy diagram.

Properties	
ID	n0
Name	Create Production Event 0
Causing Asset Key	<input type="checkbox"/> Field
Description	<input type="checkbox"/> Field
End Date	<input type="checkbox"/> Field
Headline	<input type="checkbox"/> Field
Source Production Unit	<input type="checkbox"/> Field
Start Date	<input type="checkbox"/> Field

The **Properties** window for a Create Production Event node contains the items that are described in the following table. You will use the [Field check box and subsequent options](#) to define the values in each section. The values that you define in each section will be used to populate the corresponding fields in the Production Event record that is created.

Item	Description	Notes
Causing Asset Key section	Specifies the Equipment record that represents the piece of equipment that caused the event.	The value in the Causing Asset Key section must match exactly the Entity Key of the asset that caused the event.
Description section	Specifies a detailed description of the event.	None
End Date section	Specifies the date that the event ended.	Along with the start date, the end date determines whether or not the production event will be available in the Production Event list on the Production Data Entry page.
Headline section	Specifies a short description of the event.	None
Source Production Unit section	Specifies the production unit to which loss is attributed.	The created Production Event record will be automatically linked to the specified Production Unit record.
Start Date section	Specifies the date that the event started.	Along with the end date, the start date determines whether or not the production event will be available in the Production Event list in the Losses section on the Production Data Entry page.

Note: If you add a custom field to the Production Event family, that field will also appear in the **Properties** window for the Create Production Event node.

About Specifying Dates and Times to Evaluate

You can use the Collection Filter calculation node to specify a date and time that should be used when you evaluate values. For example, consider the following example of the **Properties** window for a Collection Filter node, *Within Last Year*.

Column	Operator	Condition
Timestamp	>	Field start - 365d
Timestamp	<	Field start

In the bottom grid, you can see the columns that you can use to filter data based on dates and times.

As shown in the example, you might use these dates to indicate that you want to evaluate process historian reading values that were recorded in the past 365 days. The returned value could then be used as an input to a successor node.

You can define *specific* dates and times in the standard format *YYYY-MM-DD hh: mm: ss*. When you enter specific dates and times, the standard format will also be used to display dates and times in items that are created automatically as a result of executing policies (e.g., dates and times that are stored in Policy Recommendation records, dates and times that are displayed in email messages).

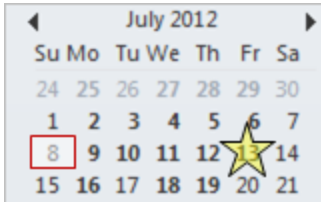
Note: With the exception of the **Policy Manager Start Page**, which may display other date formats, the standard format is used throughout the Policy Manager module regardless of any global date format settings that have been defined or the format that is defined in your Windows Region and Language settings.

Alternatively, you can specify dates and times that are *relative to the date and time at which the policy is being executed*.

For relative date formats, except for *now*, all date and time formats are evaluated against the date and time that the policy is being executed. For example, if you specify *Sunday*, the Meridium APM system will use the most recent Sunday (at 12:00:00 A.M.)

relative to the date and time that the policy is being executed. If a policy is executed on Friday, July 13, the evaluated start date would be Sunday, July 8, as illustrated in the image below, where:

- The yellow star identifies the policy execution date.
- The red box indicates the evaluated date relative to the policy execution date.



Each relative date and time format can contain the following items:

- **Constant:** Specifies a specific day or month. A [constant](#) is required.
- **Operator:** Specifies whether or not time should be added to or subtracted from the constant. You can use the + (plus sign) or - (minus sign) operators. An operator is optional.
- **Variable:** Specifies the amount of time to add to or subtract from the constant. A [variable](#) is required only if you specified an operator.

For example, consider the following start time date format:

Sunday

This format uses only the constant *Sunday* to indicate that the date range to evaluate should begin on 12:00 A.M. of the most recent Sunday, prior to the execution date.

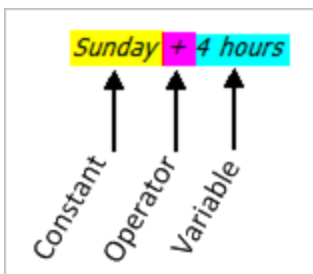
You could also indicate, however, that the date range to evaluate should begin at 4:00:00 A.M. on the most recent Sunday. To do so, you would need to add the following operator and variable to the date format:

+ 4 hours

The complete date format would be:

Sunday + 4 hours

The following image illustrates the constant, operator, and variable components of this date format:



Constants

The following table describes the acceptable constants and their meanings.

Note that:

- Where values in the **Format** column are separated by a comma, you can type *either* value to convey the same meaning.
- You must use the *exact* values that are listed in the **Format** column, (i.e., you cannot enter these values in any language other than English).

Constant	Meaning
Constants for Days	
*, start	The day and time that the policy execution began.
now	The current day and time.
t, today	12:00 A.M. of the current day.
y, yesterday	12:00 A.M. of the previous day.
Sun, Sunday	12:00 A.M. of the most recent Sunday.
Mon, Monday	12:00 A.M. of the most recent Monday.
Tue, Tuesday	12:00 A.M. of the most recent Tuesday.
Wed, Wednesday	12:00 A.M. of the most recent Wednesday.
Thu, Thursday	12:00 A.M. of the most recent Thursday.
Fri, Friday	12:00 A.M. of the most recent Friday.
Sat, Saturday	12:00 A.M. of the most recent Saturday.
Constants for Months	
Jan, January	12:00 A.M. of the most recent January 1st.
Feb, February	12:00 A.M. of the most recent February 1st.
Mar, March	12:00 A.M. of the most recent March 1st.
Apr, April	12:00 A.M. of the most recent April 1st.
May	12:00 A.M. of the most recent May 1st.
Jun, June	12:00 A.M. of the most recent June 1st.
Jul, July	12:00 A.M. of the most recent July 1st.

Constants

Constant	Meaning
Aug, August	12:00 A.M. of the most recent August 1st.
Sep, September	12:00 A.M. of the most recent September 1st.
Oct, October	12:00 A.M. of the most recent October 1st.
Nov, November	12:00 A.M. of the most recent November 1st.
Dec, December	12:00 A.M. of the most recent December 1st.

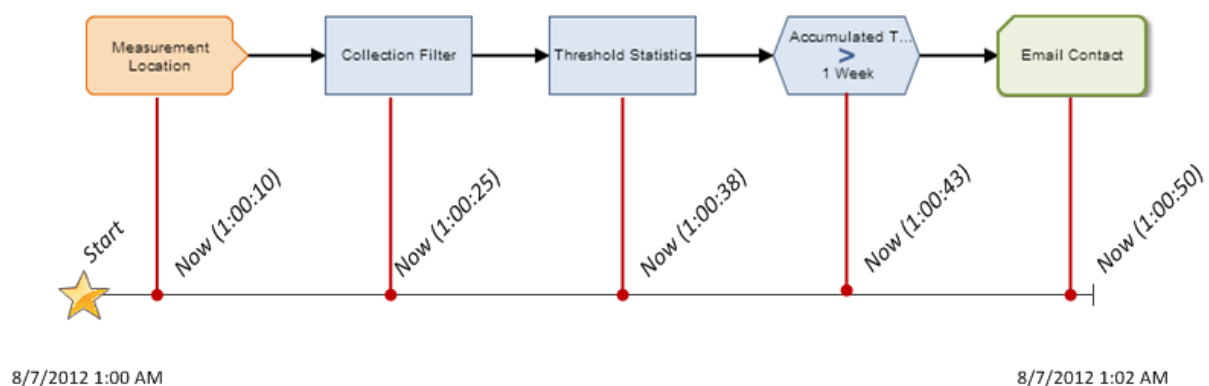
What is the Difference Between Start and Now?

When you specify dates and times that should be evaluated when the policy is executed, among other options, you can specify a time that is relative to:

- **The time at which the policy execution process began.** To do so, you would use the value *start* to indicate the start of the policy execution.
- or-
- **The time at which a specific node is executed.** To do so, you would use the value *now* to indicate the start of the node execution.

Note: These *exact* values (*start* and *now*) must be used, (i.e., you cannot enter these values in any language other than English).

For example, consider the following image, which shows a policy diagram and a time line that indicates the start of the policy execution (noted by a gold star) and the specific times at which each node is executed.



You can see from this image that the policy execution begins at 1:00 A.M. The first node, *Measurement Location*, is executed at 1:00:10 A.M. The second node, *Collection Filter*, is executed at 1:00:25 A.M., and so on.

If the *Collection Filter* node included a start time of *start - 365d*, as shown in the following image, the Meridium APM system would evaluate values beginning with 1:00:00 A.M.

What is the Difference Between Start and Now?

Properties

ID: n4

Name: Collection Filter

Collection: Field

Measurement Location: [Dropdown]

Readings: [Dropdown]

+ Filter

Column		Condition	
Timestamp	>	<input type="checkbox"/> Field start - 365d	X

Alternatively, if the Collection Filter node included a start time of *now - 365d*, as shown in the following image, the Meridium APM system would evaluate values beginning with 1:00:25 A.M.

Properties

ID: n4

Name: Collection Filter

Collection: Field

Measurement Location: [Dropdown]

Readings: [Dropdown]

+ Filter

Column		Condition	
Timestamp	>	<input type="checkbox"/> Field now - 365d	X

Variables

The following table describes the acceptable variables and their meanings.

Note that:

- Where values in the **Variables** column are separated by a comma, you can type *either* value to convey the same meaning.
- You must use the *exact* values that are listed in the **Variables** column, (i.e., you cannot enter these values in any language other than English).

Variables	Meaning
s, sec, second, seconds	Adjusts the time by the specified number of seconds (e.g., 7 seconds).
m, min, minute, minutes	Adjusts the time by the specified number of minutes (e.g., 10 minutes).
h, hour, hours	Adjusts the time by the specified number of hours (e.g., 5 hours).
d, day, days	Adjusts the time by the specified number of days (e.g., 30 days).
mo, month, months	Adjusts the time by the specified number of months (e.g., 6 months).
y, year, years	Adjusts the time by the specified number of years (e.g., 2 years).
w, week, weeks	Adjusts the time by the specified number of weeks (e.g., 3 weeks).
[ddd].HH:MM:SS	Adjusts the time by the specified time period (e.g., 15.12:15:35).

Note: You can use decimals for any variable *except* for those specifying days and months.

About Specifying an Amount of Time to Evaluate

Some node properties allow you to specify an amount of time that should be used when evaluating values. For example, consider the following **Properties** window for a *Greater Than Condition* node.

Note: The yellow highlighting has been added to the image to highlight the text box into which you could specify an amount of time.

The screenshot shows a 'Properties' window with the following elements:

- ID: n2
- Field
- Threshold Statistics 1 (dropdown)
- Accumulated Time (dropdown)
- Display: Field (dropdown)
- A blue arrow pointing right
- Field
- A yellow highlighted text box below the second 'Field' checkbox.

Below the **Field** check box, you can see the text box that allows you to type a value. You might use this value to indicate, for example, that you want to evaluate reading values that were recorded in the past month and perform some action if the reading values exceeded 100 for a total of *one week* (not necessarily one week *straight*).

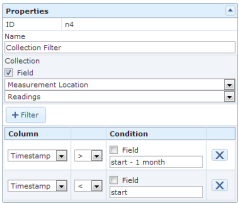
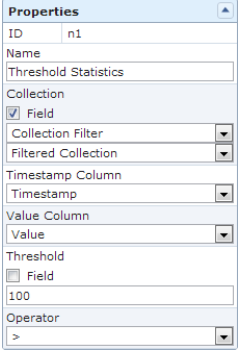
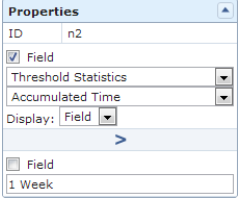
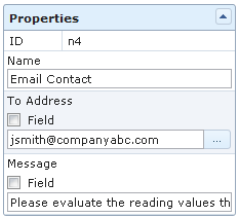
To further illustrate this example, consider the following policy diagram and node properties.

Policy Diagram



Node Properties

Node	Properties Window	Description
Measurement Location		Represents the Measurement Location family.

Node	Properties Window	Description
Collection Filter		<p>Filters the reading values to only the values which have start times within the last month. The notations <i>start - 1month</i> and <i>start</i> are used to determine the specific date range that should be used when determining which reading values to include in the evaluation process.</p>
Threshold Statistics		<p>Evaluates the reading values (associated with the Measurement Location that will be identified by a policy instance) to determine if any of them are greater than 100.</p>
Accumulated Time > 1 week		<p>Indicates that if reading values greater than 100 were recorded for a total of one week or more in the past month, the Meridium APM system should perform the action identified by the successor node <i>Email Contact</i>.</p> <div style="border: 1px solid yellow; padding: 5px; margin-top: 10px;"> <p>Note: The amount of time identified by the notation <i>1week</i> is used to determine the duration against which the returned reading values (i.e., the values that were returned from the threshold comparison) are compared.</p> </div>
Email Contact		<p>Sends an email message to jsmith@companyabc.com.</p>

When specifying an amount of time, you can use any of the following formats.

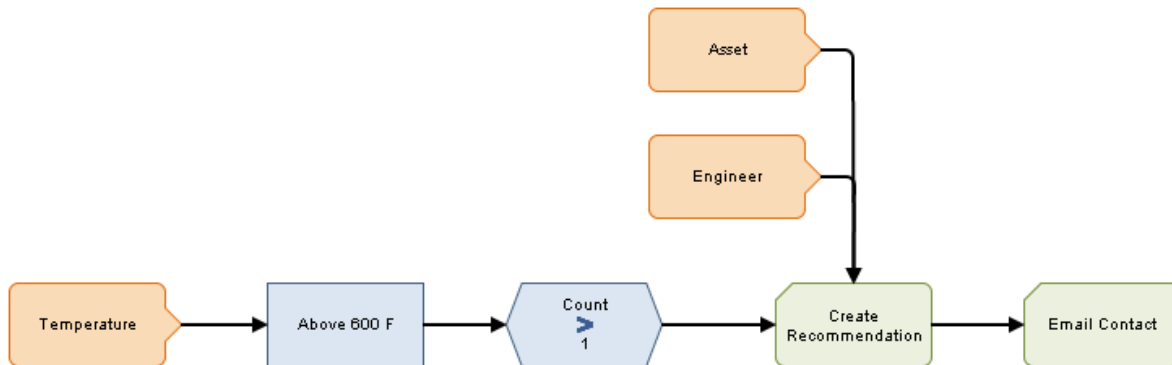
Format	Description	Example
<n> s, sec, second, seconds	Specifies a number of seconds, where <n> is the desired number.	15 seconds
<n> m, min, minute, minutes	Specifies a number of minutes, where <n> is the desired number.	10 minutes
<n> h, hour, hours	Specifies a number of hours, where <n> is the desired number.	3 hours
<n> d, day, days	Specifies a number of days, where <n> is the desired number.	5 days
<n> w, week, weeks	Specifies a number of weeks, where <n> is the desired number.	7 weeks
d.hh:mm:ss	Specifies a number of days, hours, minutes, and seconds.	300.23:13:22 (300 days, 23 hours, 13 minutes, and 22 seconds)

Note the following guidelines and limitations of time span formats:

- You can use decimals for any format *except* for those specifying days.
- You can use negative numbers if needed.
- You cannot specify a number of months (e.g., *1 month*) because some months have a different number of days (e.g., July has 31 days, and September has 30 days). Instead, you can specify a number of days (e.g., *30 days*).
- You cannot specify a number of years (e.g., *1 year*) because some years have a different number of days. For example, leap years contain 366 days, and non-leap years contain 365 days.
- You must use the *exact* values that are listed in the **Format** column, (i.e., you cannot enter these values in any language other than English).

Example Diagram: Monitoring a Single Value

The following image shows the policy diagram for [the single value monitoring example](#).

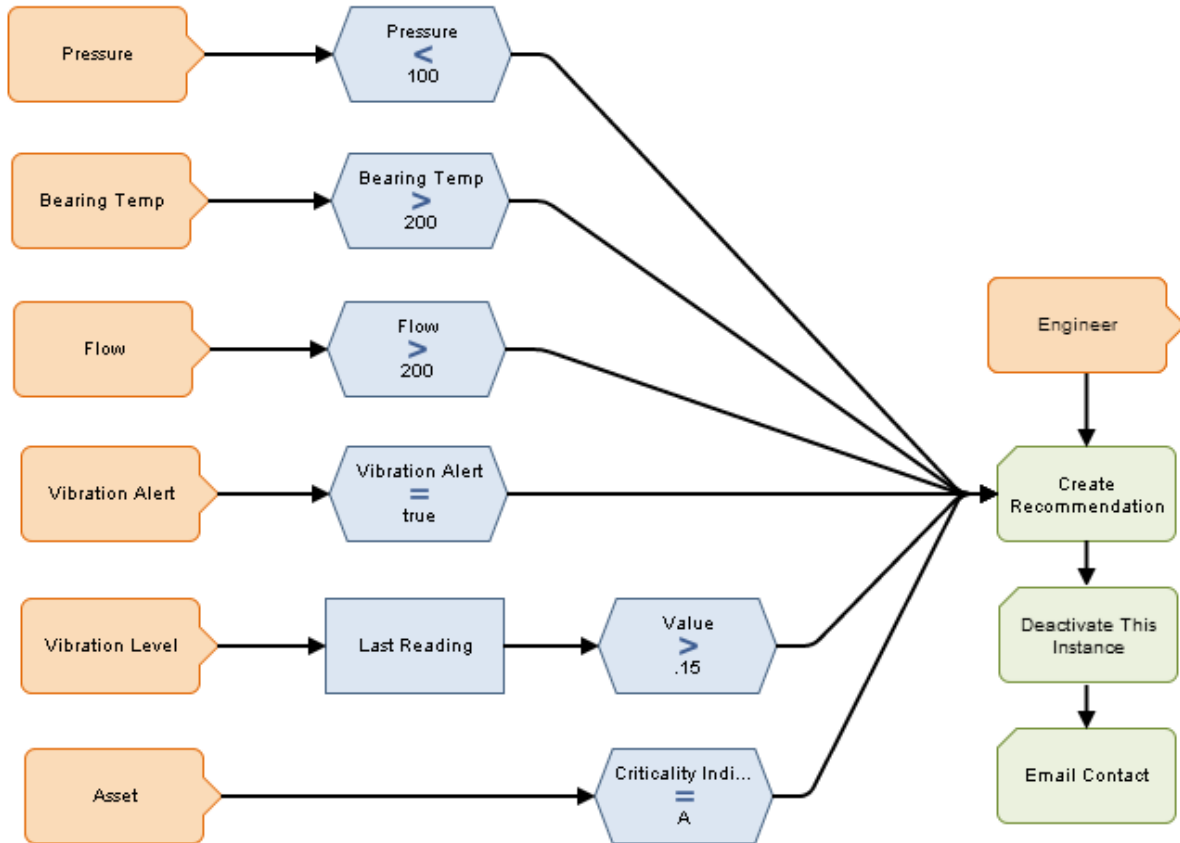


This policy diagram contains the following nodes.

Node Name	Node Type	Description
Temperature	Xi Tag	Provides reading values that are passed to the <i>Above 600 F</i> node.
Above 600 F	Threshold Statistics	Compares the reading values returned by the <i>Temperature</i> node to the threshold value <i>600</i> . If any reading values are greater than or equal to 600, the <i>Count > 1</i> node is evaluated.
Count > 1	Condition node	Compares the <i>number</i> of times the reading values were greater than or equal to 600 to <i>1</i> .
Create Recommendation	Create Recommendation	Creates a Policy Recommendation record.
Email Contact	Email Contact	Sends an email message.
Engineer	User	Supplies a User ID and email address to the <i>Email Contact</i> node.
Asset	Entity	Represents the Equipment family and supplies a value from the Equipment ID field to the Policy Recommendation record. When the policy is executed, because the Asset ID field in the Policy Recommendation record is populated with the value in the Equipment ID field in an Equipment record, the Policy Recommendation record will be linked automatically to that Equipment record.

Example Diagram: Monitoring Multiple Values

The following image shows the policy diagram for a [policy that monitors multiple values](#).



This policy diagram contains the following nodes:

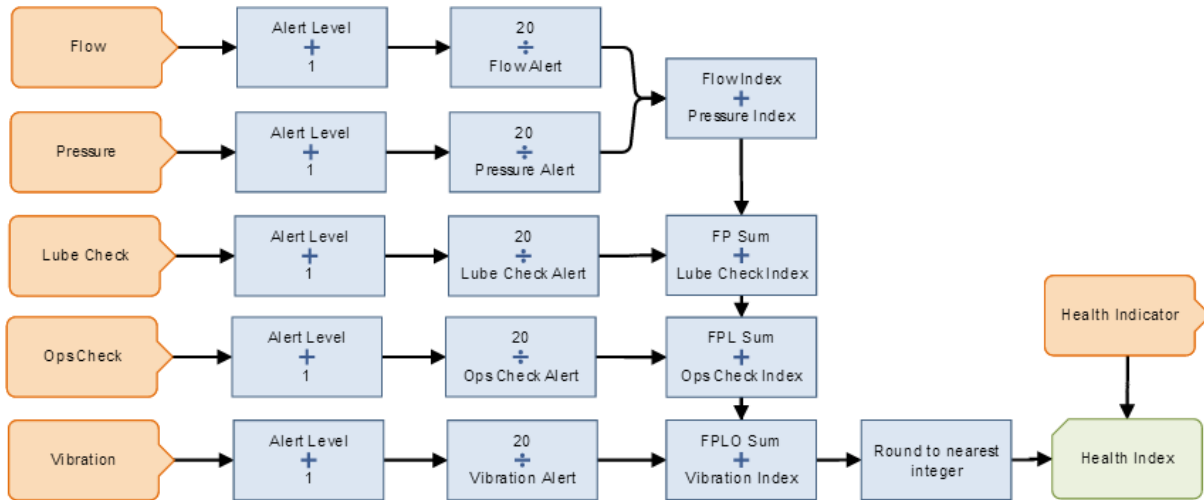
Node Name	Node Type	Description
Pressure	Xi Tag	Provides DA Reading Values that are evaluated by the the <i>Pressure < 100</i> node.
Bearing Temp	Xi Tag	Provides DA Reading Values that are evaluated by the the <i>Bearing Temp > 200</i> node.
Flow	Xi Tag	Provides DA Reading Values that are evaluated by the the <i>Flow > 200</i> node.
Vibration Alert	AMS Asset	Provides information about alerts related to AMS Asset records. This information is evaluated by the <i>Vibration Alert = True</i> node.

Example Diagram: Monitoring Multiple Values

Node Name	Node Type	Description
Vibration Level	Measurement Location	Provides Measurement Location readings that are evaluated by the <i>Last Reading</i> and <i>Value > .15</i> nodes.
Asset	Entity	Represents the Equipment family and supplies a criticality indicator to the Condition node <i>Criticality Indicator = A</i> .
Pressure < 100	Condition	Evaluates the process historian pressure readings to determine if any readings are less than 100.
Bearing Temp > 200	Condition	Evaluates the process historian temperature readings to determine if any readings are greater than 200.
Flow > 200	Condition	Evaluates the process historian flow readings to determine if any readings are greater than 200.
Vibration Alert = True	Condition	Evaluates AMS Asset records to determine if any are linked to AMS Asset Alert records with the value <i>True</i> in the Active field.
Last Reading	Last	Evaluates the collection of readings associated with the <i>Vibration Level</i> node and returns the last row in the collection, which, in this case, contains the most recent reading.
Value > .15	Condition	Evaluates the value in the Value field of the Measurement Location reading returned by the <i>Last Reading</i> node to determine if it is greater than .15.
Criticality Indicator = A	Condition	Evaluates the value in the Criticality Indicator field in the Equipment record to determine if it contains the value <i>A</i> .
Engineer	User	Supplies user information to the <i>Create Recommendation</i> and <i>Email Contact</i> nodes.
Create Recommendation	Create Recommendation	Creates a Policy Recommendation record if <i>all</i> of the preceding conditions are met.
Deactivate This Instance	Deactivate This Instance	Automatically deactivates the policy instance whose values caused the Deactivate This Instance node to be executed.
Email Contact	Email Contact	Sends an email message.

Example Diagram: Creating Health Indicator and Health Indicator Value Records

The following image shows the policy diagram for [the example policy that creates a Health Indicator record and a Health Indicator Value record](#).



This policy diagram contains the following nodes.

Node Name	Node Type	Description
Flow, Pressure, Lube Check, Ops Check, and Vibration	Health Indicator	Represents various existing Health Indicator records whose primary source records are Measurement Locations. These records provide the values used in successor nodes to determine the overall health of the pump.
Alert Level + 1	Add	Adds 1 to the value in the Alert Level field in the corresponding Health Indicator record. The resulting value (n+1) is used as the denominator in the successor Divide nodes.

Node Name	Node Type	Description
20 / <Health Indicator name> Alert	Division	Divides 20 by the result of the corresponding Add node. This value represents the Health Index contribution for each node. 20 is used as the numerator to ensure that each individual Health Indicator has equal weight, (i.e., since there are five Health Indicators, 20 is one-fifth of 100).
Various	Add	Combine the results from the Division nodes. This sum represents the overall health index of the pump on a numeric scale between 1 and 100, where a higher value is better than a lower value.
Round to nearest integer	Round	Rounds the value of the overall health index to the nearest integer.
Health Index	Add Value to Health Indicator	Adds the rounded value of the overall health index to a new Health Indicator Value record and links the record to a Health Indicator record.
Health Indicator	Health Indicator	Represents the Health Indicator record to which the Health Indicator Value record will be linked when the policy is executed.

Continuing with the example, consider that the following Health Indicator records contain the following values in the Alert Level field:

- Flow Rate: 0
- Pressure: 1
- Lubrication: 0
- General Operation: 1
- Vibration: 1

When the policy is executed:

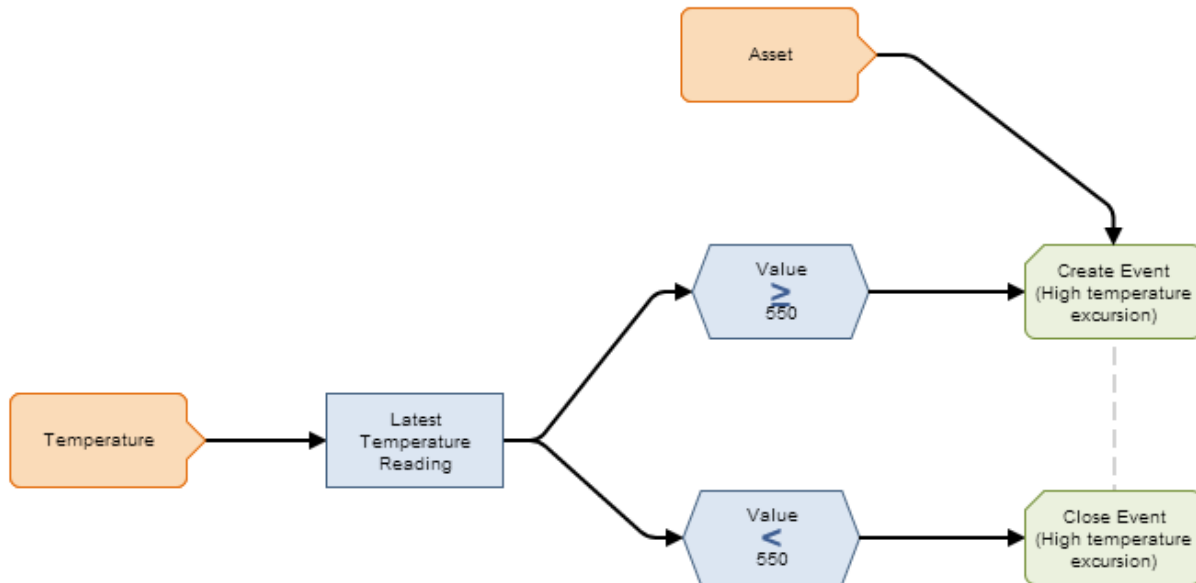
- The Calculation nodes that are described in step 2 will yield the following values for the Health Indicators:
 - Flow Rate: 20 (i.e., $20/[0+1]=20$)
 - Pressure: 10 (i.e., $20/[1+1]=10$)
 - Lubrication: 20 (i.e., $20/[0+1]=20$)

Example Diagram: Creating Health Indicator and Health Indicator Value Records

- General Operation: 10 (i.e., $20/[1+1]=10$)
- Vibration: 10 (i.e., $20/[1+1]=10$)
- The Calculation nodes that are described in step 3 will yield a sum of 70 (i.e., $20+10+20+10+10=70$)
- A new Health Indicator record, [as specified in the policy instance associated with the Health Indicator node](#), will be created automatically that indicates the overall health of the pump.
- A Health Indicator Value record associated with the new Health Indicator record will be created automatically. The Value (Numeric) field will be populated with the value 70.
- You can view the new Health Indicator record in AHM just like you can view Health Indicator records whose primary sources are Measurement Location, Xi Tag, or KPI records.

Example Diagram: Creating and Updating a Policy Event Record

The following image shows the policy diagram for the [creating and updating a Policy Event record example](#).



The policy model diagram the following nodes.

Node Name	Node Type	Description
Temperature	Xi Tag	Provides reading values that are passed to the <i>Latest Temperature Reading</i> node.
Latest Temperature Reading	Last	Provides reading values that are passed to the <i>Value ≥ 550</i> and <i>Value < 550</i> nodes for evaluation.
Value ≥ 550	Condition node	Compares the reading value returned by the <i>Latest Temperature Reading</i> node. If the reading value is greater than or equal to 550, the Create Event node is triggered.
Value < 550	Condition node	Compares the reading value returned by the Latest Temperature Reading node. If the reading value is less than 550, the Close Event node is triggered.

Example Diagram: Creating and Updating a Policy Event Record

Node Name	Node Type	Description
Asset	Entity	Represents the asset that associated with the Policy Event record that is created by the Create Event node.
Create Event (High temperature excursion)	Create Event	Creates a Policy Event record.
Close Event (High temperature excursion)	Close Event	Updates the Policy Event record to indicate that the event is closed.

Primary Records and Primary Nodes

In many cases, you will create policies that monitor a lot of different information about a specific item, such as a piece of equipment. In these cases, the policy will contain an Input node to identify the piece of equipment and multiple, additional Input nodes, where each additional Input node represents a specific piece of information that you want to monitor about the piece of equipment. For each of these additional Input nodes, within a policy instance, you will need to assign a specific record to the node.

For example, consider a policy that monitors pressure, temperature, and failure cost for a specific piece of equipment. In other words, the policy includes the following nodes:

- Equipment
- Pressure
- Temperature
- Failure Cost

You would assign a specific Equipment record to the *Equipment* node to identify the piece of equipment that you want to monitor. Then, to the *Pressure* node, you might assign a Measurement Location record because the Measurement Location record and the Readings to which it is linked store pressure readings for the piece of equipment. Likewise, for the *Failure Cost* node, you might map a Work History record to it because the Work History record stores the failure cost for the piece of equipment.

When mapping the Measurement Location and Work History records to their associated Input nodes, you can define the search criteria manually so that it will return only Measurement Location and Work History records that are linked to the Equipment record representing the piece of equipment you are monitoring. Alternatively, you can configure the policy such that the search criteria will be defined in this way *automatically*. To do so, you can identify the Equipment record as the *primary record*. After identifying the Equipment record as the primary record, when you perform a search for the record that you want to assign to another Input node, the search will be filtered automatically to return only records that are linked to the primary record (the Equipment record).

Throughout this documentation, we use the following terms regarding this functionality:

- **Primary record:** The record that is being used to filter the search results when searching for records to assign to nodes.
- **Primary node:** The node with which the primary record is associated.

In [policies that contain a primary node and primary record](#), a visual indicator (a yellow star) appears in the following locations:

- In the policy diagram, in the upper left corner of the primary node, as shown in the following image.



- In the policy instance, in the **Source** column in the row representing the primary record, as shown in the following image.

Source	Instance Value
★ Equipment	~ #1 STEAM REHEATER ~ HXST 130

The previous example and explanation assume that the policy instance does *not* contain any rows representing records that are *not* assigned to an Input node. In other words, it assumes that each row in the policy instance represents a record that *is* assigned to a node in the policy diagram. In some cases, a policy instance might contain a row representing a record that is *not* assigned to a node in the policy diagram. [In these cases, you can still specify this record as the primary record](#), but there will *not* be a primary node with which to associate it.

Workflow for Identifying Primary Records

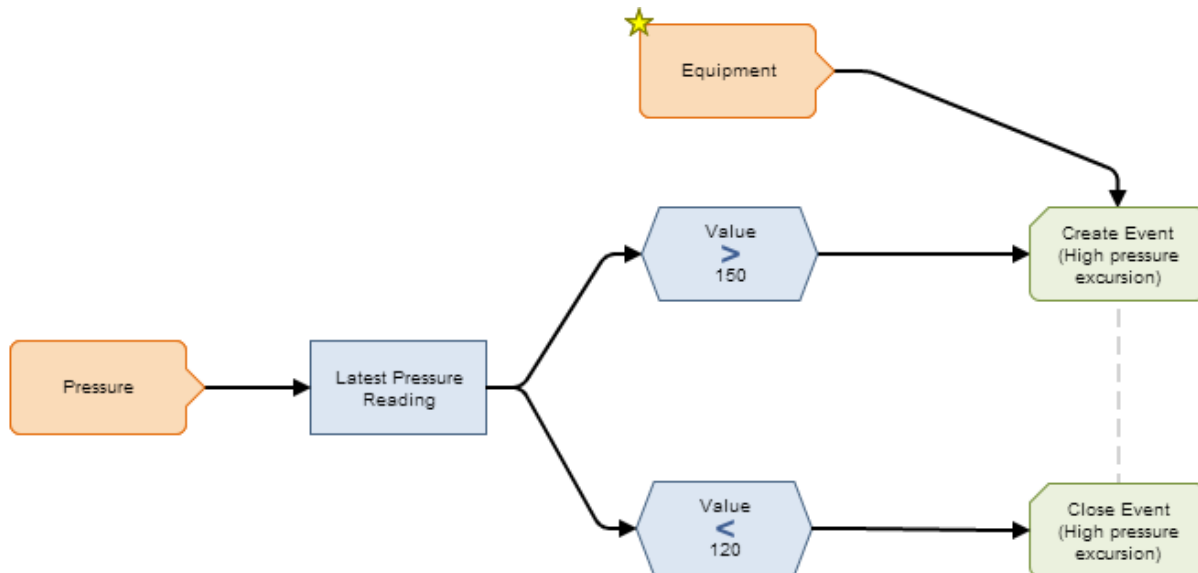
To identify a primary record, you will need to complete the following steps:

1. In the policy diagram, [identify the primary node](#) (i.e., the node that will be associated with the primary record). This step is *optional*. It must be completed only if you want the [primary record to be associated with a node in the diagram](#).
2. In the policy instance, specify the primary record. This step is *required* regardless of whether or not you have identified the primary node.

Example: Primary Node and Primary Record

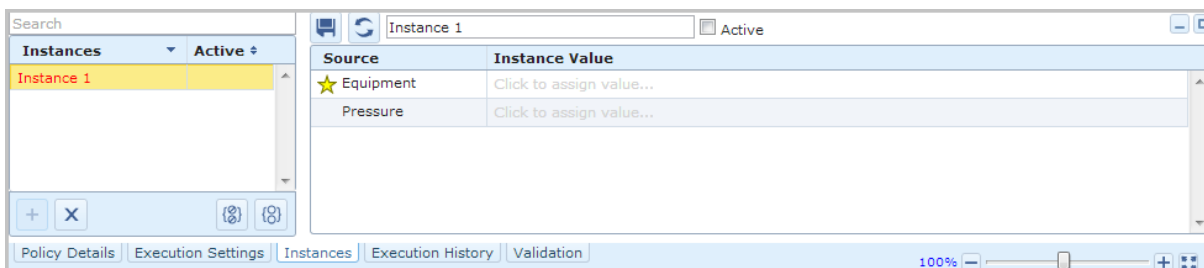
The following example illustrates a policy in which a primary node and primary record have been identified.

Consider the following image of a policy that monitors the pressure readings recorded for a particular piece of equipment.



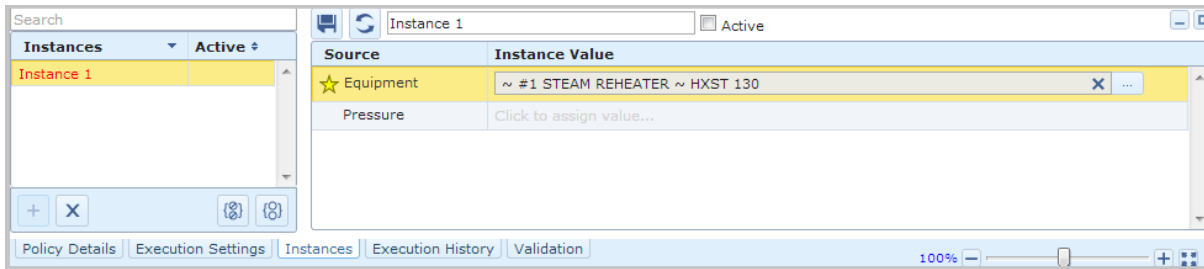
The *Equipment* node has been identified as the primary node, as indicated by the yellow star that appears on the top left corner of the node in the diagram.

Continuing with the example, before you can execute this policy, policy instances must be created, and records must be [assigned to the Input nodes](#), *Equipment* and *Pressure*, as illustrated in the following image of the **Instances** tab.

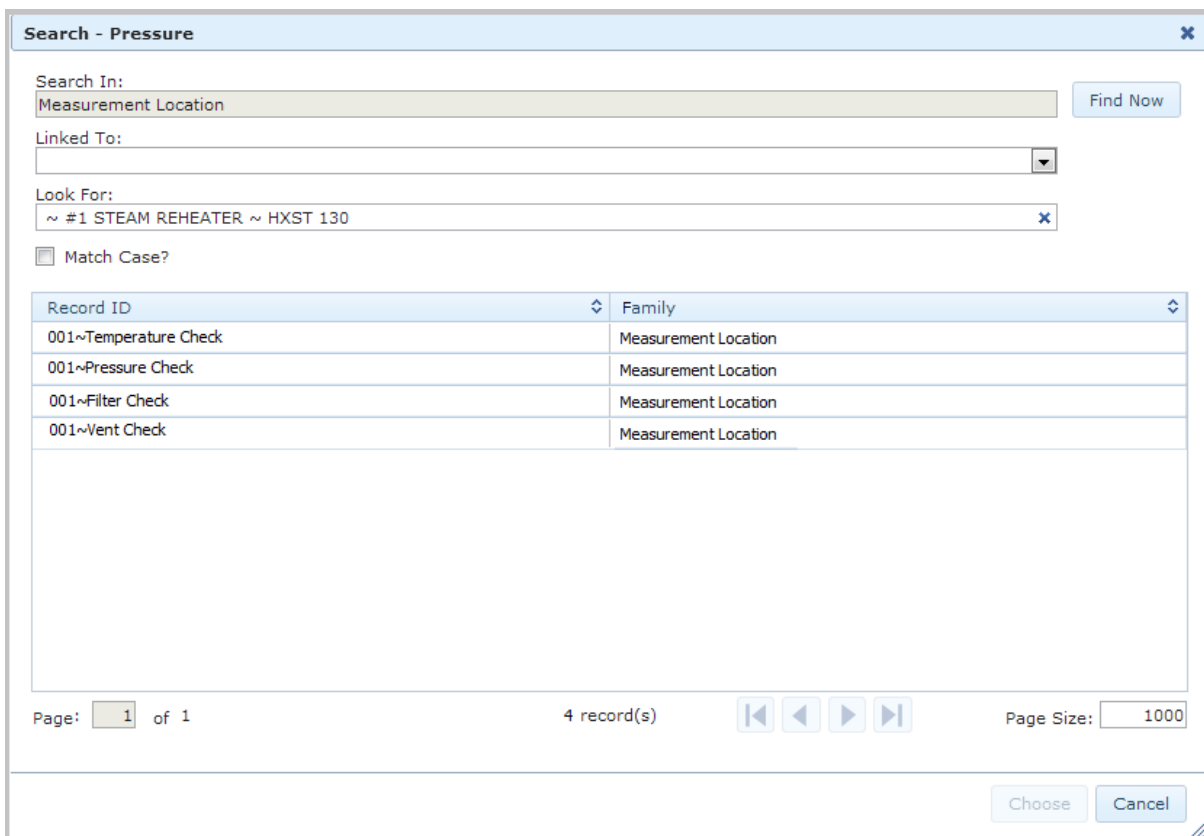


Because you have specified the *Equipment* node as the primary node, a yellow star appears to the left of the node name in the **Source** column. For this node, you will need to [specify the primary record](#), (i.e., the *Equipment* record). In the following image, the *Equipment* record with the Record ID ~ #1 STEAM REHEATER ~ HXST 130 has been specified.

Example: Primary Node and Primary Record



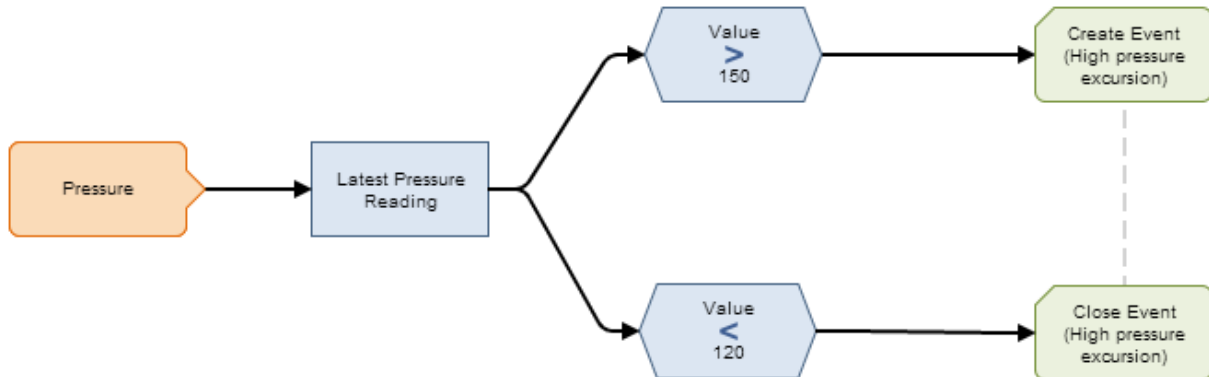
For the Pressure node, you will need to specify the associated record that stores the pressure readings that you want to monitor for the piece of equipment, (i.e., the Measurement Location record that is linked to the Equipment record that has been specified as the primary record). Because you have specified the Equipment record with the Record ID ~ #1 STEAM REHEATER ~ HXST 130 as the primary record, the process for assigning records to the Pressure node is streamlined in that the search criteria defined by default on the **Search<Node>** window, where **<Node>** is the name of the policy node for which you accessed the window, will return only Measurement Location records that are linked to the Equipment Record ~ #1 STEAM REHEATER ~ HXST 130, as shown in the following image.



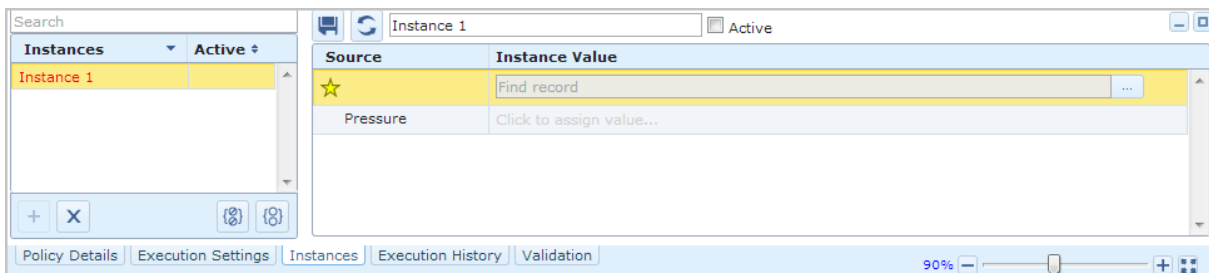
Example: Primary Record Only

The following example illustrates a policy in which a primary record has been identified, but a primary node has *not* been identified.

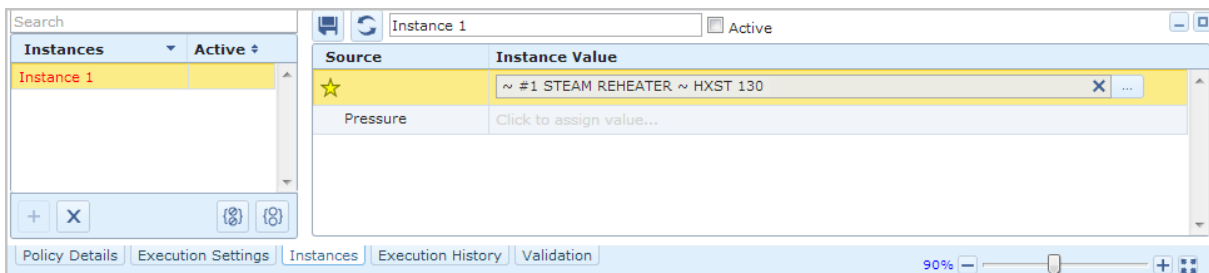
For example, consider that you use AHM to analyze the piece of equipment represented by the Equipment record with the Record ID ~ #1 STEAM REHEATER ~ HXST 130. Additionally, you have created the policy shown in the following image to monitor the pressure readings that are recorded for this piece of equipment.



Before you can execute this policy, [policy instances must be created](#), and a record must be [assigned to the Input node](#), *Pressure*, as illustrated in the following image of the **Instances** tab.



In the grid in the [Details area](#), in the row associated with the primary record (the row containing a yellow star in the **Source** column), you will need to specify the Equipment record that you analyze in AHM, (i.e., the Equipment record with the Record ID ~ #1 STEAM REHEATER ~ HXST 130) as shown in the following image.



Example: Primary Record Only

For the Pressure node, you will need to specify the associated record that stores the pressure readings that you want to monitor for the piece of equipment, (i.e., the Measurement Location record that is linked to the Equipment record that has been specified as the primary record). Because you have specified the Equipment record with the Record ID ~ #1 STEAM REHEATER ~ HXST 130 as the primary record, the process for assigning records to the Pressure node is streamlined in that the search criteria defined by default on the **Search<Node>** window, where **<Node>** is the name of the policy node for which you accessed the window, will return only Measurement Location records that are linked to the Equipment record ~ #1 STEAM REHEATER ~ HXST 130, as shown in the following image.

The screenshot shows a search window titled "Search - Pressure". The search criteria are as follows:

- Search In:** Measurement Location
- Linked To:** (empty dropdown)
- Look For:** ~ #1 STEAM REHEATER ~ HXST 130
- Match Case?

Record ID	Family
001~Temperature Check	Measurement Location
001~Pressure Check	Measurement Location
001~Filter Check	Measurement Location
001~Vent Check	Measurement Location

Page: 1 of 1 4 record(s) Page Size: 1000

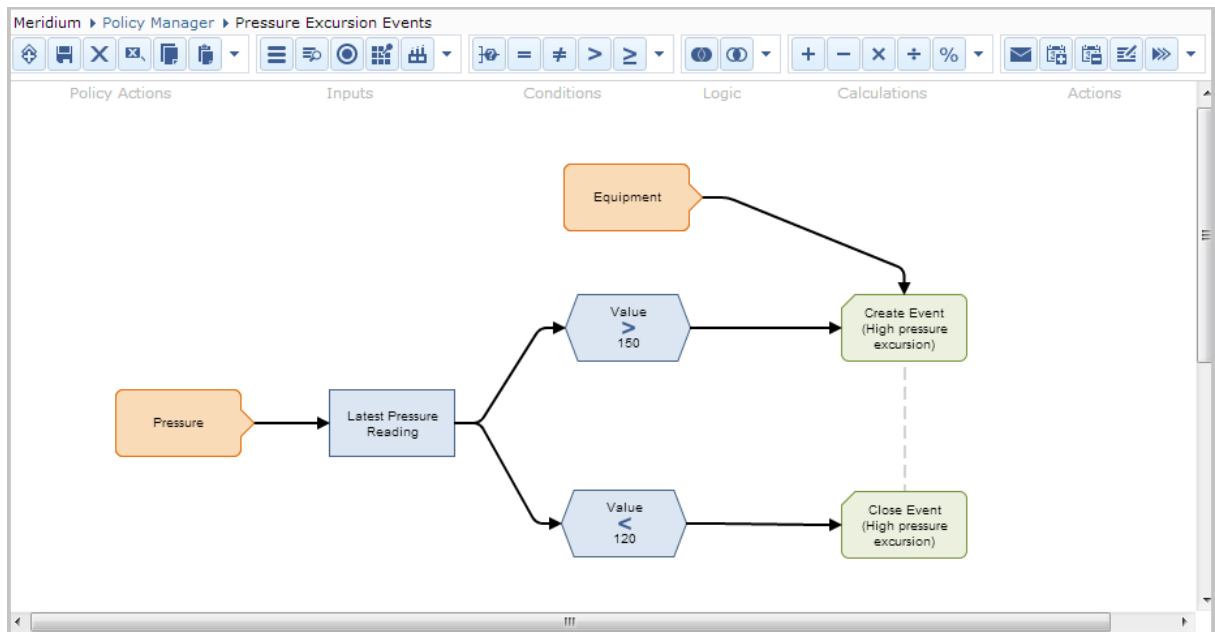
Buttons: Choose, Cancel

Identifying the Primary Node in a Policy

If the policy [includes an Input node that will be associated with the primary record](#), you can specify that node as the primary node. When you do this, a yellow star will be added to the upper left corner of the node in the policy diagram to indicate the node is associated with the primary record. You can identify only one primary node in a policy. Any Input node except for a [Constant](#), [Query](#), or [Point Value](#) node can be identified as the primary node. After you identify the primary node, you must [specify the primary record itself](#) in order to streamline the process for assigning records to other nodes.

To identify the primary node in a policy:

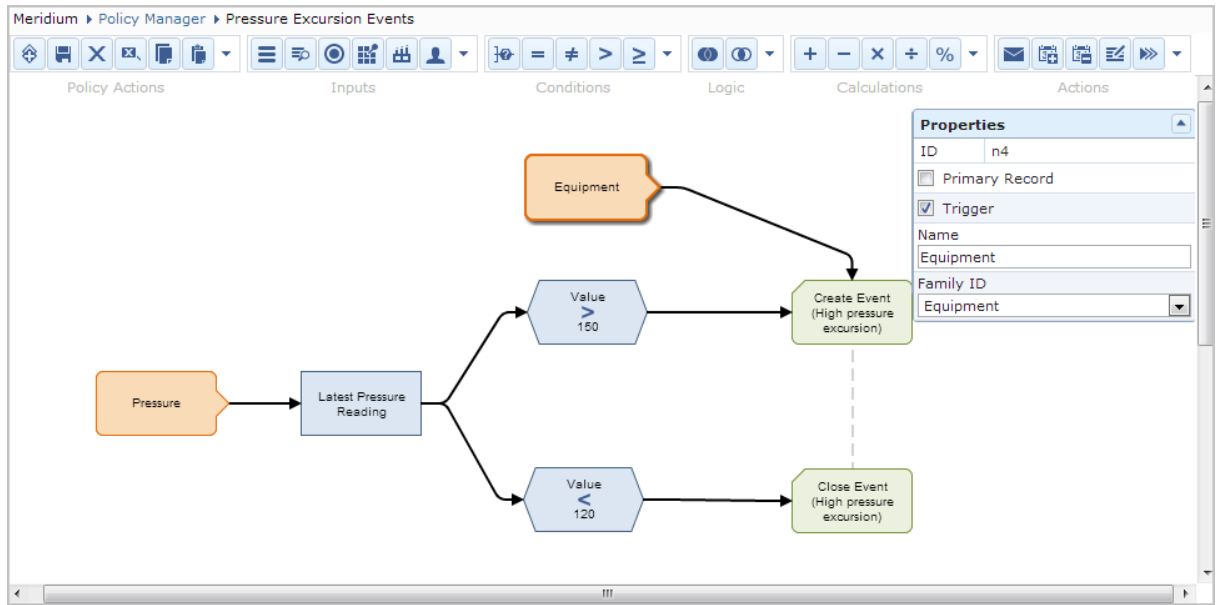
1. Access the **Policy** page for the policy for which you want to identify a primary node. The following image shows an example of the policy diagram for the policy *Pressure Excursion Events*.



2. In the policy diagram, select the Input node that you want to specify as the primary node.

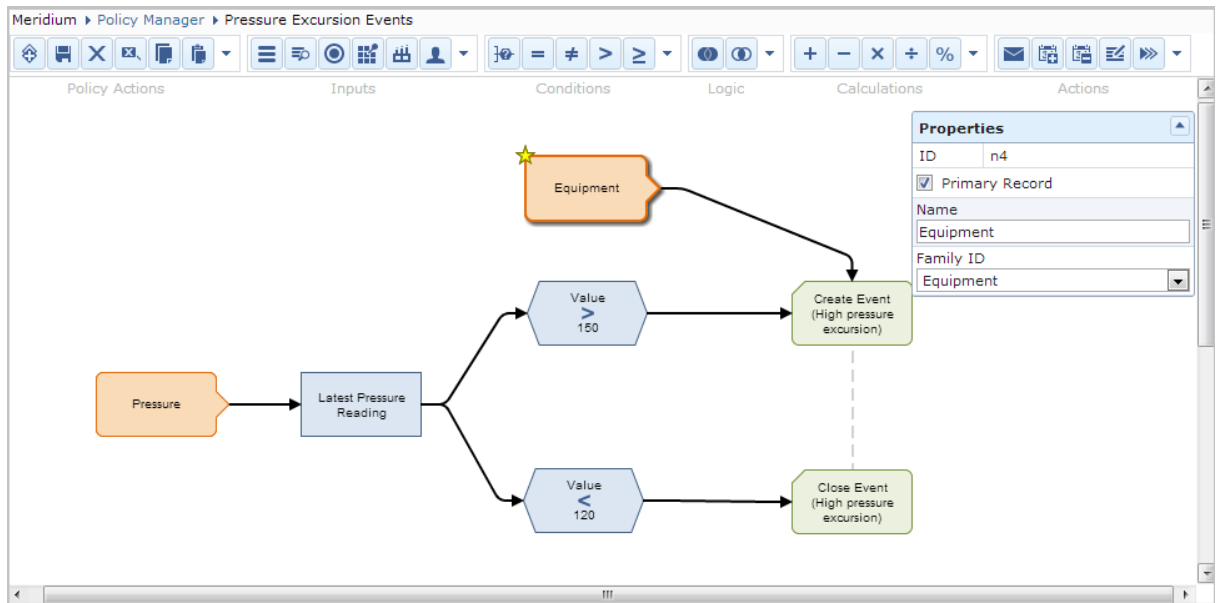
The **Properties** window appears in the policy diagram. In the following image, the *Equipment* node is selected.

Identifying the Primary Node in a Policy



3. On the **Properties** window, select the **Primary Record** check box.

A yellow star appears on the top left corner of the node in the policy diagram, indicating that the node is the primary node in the policy.



At this point, before you can execute the policy, you must:

- [Create at least one policy instance.](#)
- Within each policy instance that you create, [specify the primary record that is associated with the primary node.](#) Doing so will streamline the process for assigning records to other nodes in the policy diagram.

Specifying the Primary Record in a Policy when a Primary Node is Identified

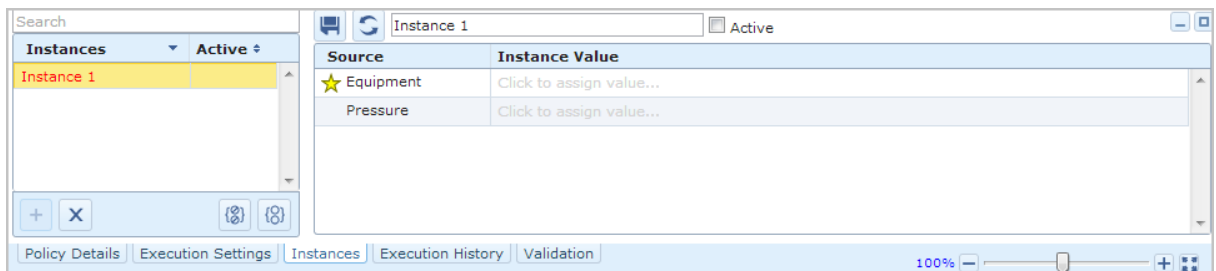
When a [primary node has been identified in the policy](#), you can specify the primary record in order to streamline the process for [assigning records to other nodes](#). You can also [specify a primary record when a primary node has not been identified in the policy](#).

The following instructions assume that one or more [instances have been created](#) for the policy for which you want to specify the primary record.

To specify the primary record in a policy when a primary node is identified:

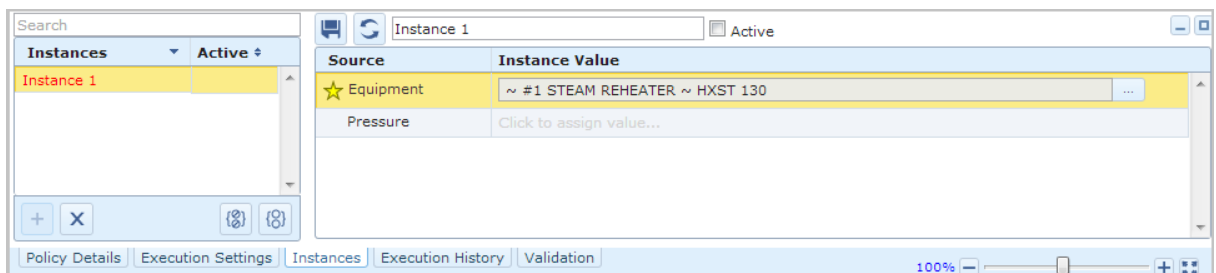
1. On the **Policy** page for the policy for which you want to specify a primary record, access the **Instances** tab.

A yellow star appears to the left of the primary node name in the grid in the [Details area](#), as shown in the following image.



2. [Assign the primary record to the primary node.](#)

The Record ID of the selected record appears in the **Instance Value** column. The following image shows an example where the Equipment record with the Record ID *~ #1 STEAM REHEATER ~ HXST 130* has been specified as the primary record, which is associated with the primary node *Equipment*.



At this point, you can continue [assigning records to other nodes](#) in the policy instance. As a result of specifying the primary record, when you use the **Search <Node>** window, where **<Node>** is the name of the policy node for which you accessed the window, to select records to assign to other nodes, the search criteria that is defined by default will return only records that are linked to the primary record. Continuing with the example used in these instructions, the

search criteria that is defined by default will return only records that are linked to the Equipment record with the Record ID ~ #1 *STEAM REHEATER* ~ *HXST130*.

Specifying the Primary Record in a Policy when a Primary Node is Not Identified

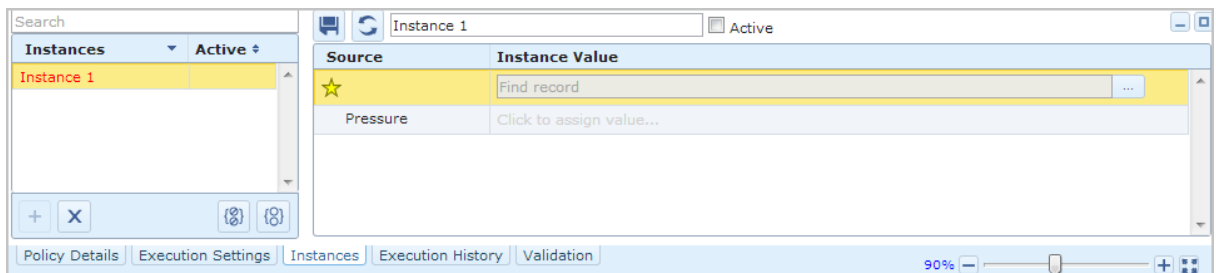
When a [primary node has not been identified in the policy](#), you can specify the primary record in order to streamline the process for [assigning records to other nodes](#). You can also [specify the primary record when a primary node has been identified](#) in the policy.

The following instructions assume that one or more [instances have been created](#) for the policy for which you want to specify the primary record.

To specify the primary record in a policy when a primary node is not identified:

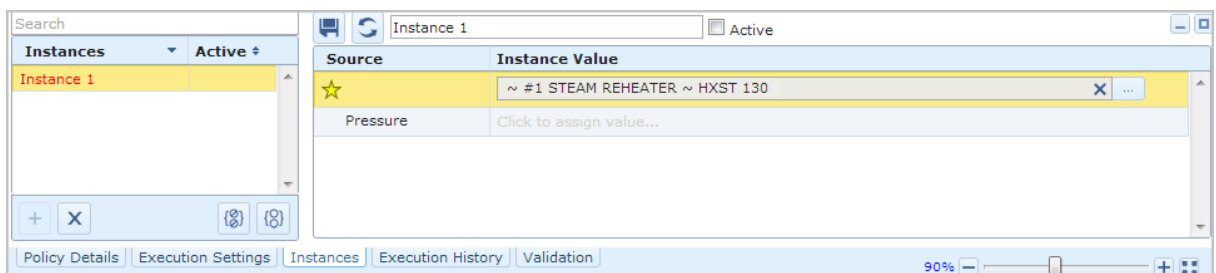
1. On the **Policy** page for the policy for which you want to specify a primary record, access the **Instances** tab.

An empty row appears in the grid in the [Details area](#), displaying a yellow star in the **Source** column, as shown in the following image.



2. In the empty row with the yellow star in the **Source** column, click in the **Instance Value** cell to access the **Search <Node>** window, where **<Node>** is the name of the policy node for which you accessed the window.
3. Use the **Search <Node>** window to select the primary record.

The Record ID of the selected record appears in the **Instance Value** column. The following image shows an example where the Equipment record with the Record ID *~ #1 STEAM REHEATER ~ HXST130* has been specified as the primary record, which is *not* associated with a primary node.



At this point, you can continue [assigning records to other nodes](#) in the policy instance. As a result of specifying the primary record, when you use the **Search <Node>** window to select records to assign to other nodes, the search criteria that is defined by default will return only records that are linked to the primary record.


Continuing with the example used in these instructions, the search criteria that is defined by default will return only records that are linked to the Equipment record with the Record ID ~ *#1 STEAM REHEATER ~ HXST130*.

Creating Policy Instances

Policy instances represent the records whose values a policy monitors. Policy instances will be associated automatically with the policy that you are viewing when you create the policy instances. In other words, policy instances cannot be created outside of the context of the policy with which they are associated. After you create policy instances, before you can execute a policy, you must [assign specific records to policy nodes](#).

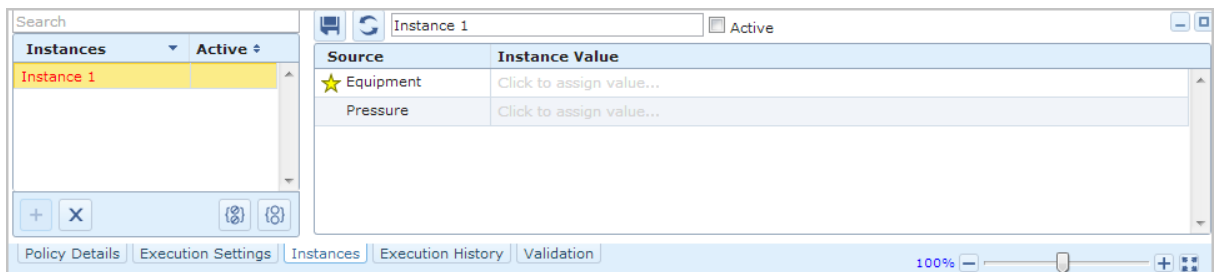
Note: Policy instances can be created only if you are viewing a [saved](#) policy, and you must be a member of the MI Policy Designer Security Group or the MI Policy User Security Group in order to create a policy instance.

To create a policy instance:

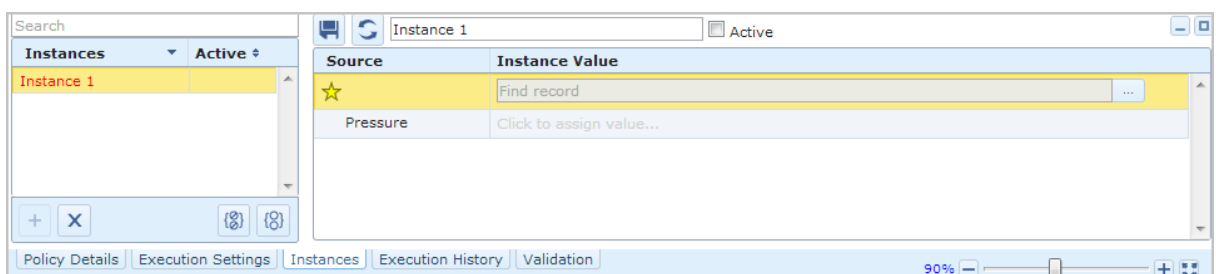
1. [Open the policy](#) for which you want to create a new policy instance.
2. On the [Instances tab](#), click the  button.

A new policy instance appears in the grid in the [Search area](#), and a grid appears in the [Details area](#) displaying one row for each node in the policy that requires [values to be mapped from a specific record](#). If a primary node has been identified in the policy diagram, a yellow star appears in the **Source** column to the left of that node name. If a primary node has not been identified in the policy diagram, the grid also contains one additional, empty row to represent the primary record that you will later define.

The following image shows an example of the **Instances** tab when a primary node (*Equipment*) has been identified.



The following image shows an example of the **Instances** tab when a primary node has *not* been identified.



At this point, you must [assign specific records to the policy nodes](#).

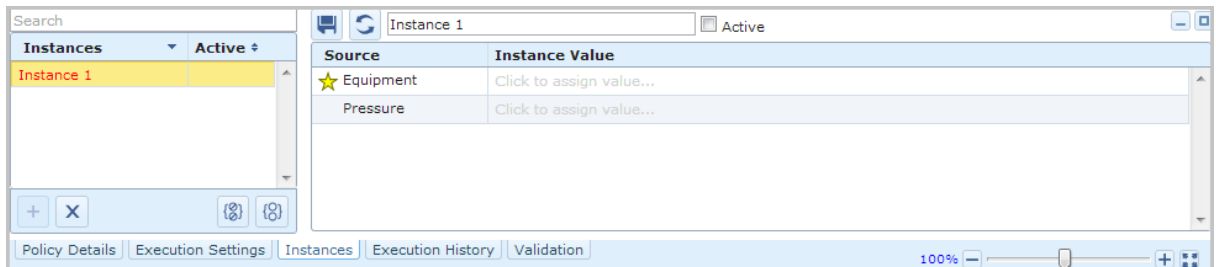
Assigning Specific Records to Nodes

Before you can [execute a policy](#), you must assign records to each Input node in the policy with the exception of Query and Constant nodes. If a primary node has been identified in the policy diagram, the record that you assign to that node will become the [primary record](#). If a primary node has [not been identified](#) in the policy diagram, you can optionally choose whether or not to specify a primary record that will not be associated with a policy node.

Note that for Point Value nodes, you must assign a specific record *and* a specific field. You can use the following instructions to assign the record. To assign the field, in the grid in the [Details area](#), you can use the list that appears below the Record ID of the selected record.

To assign specific records to nodes:

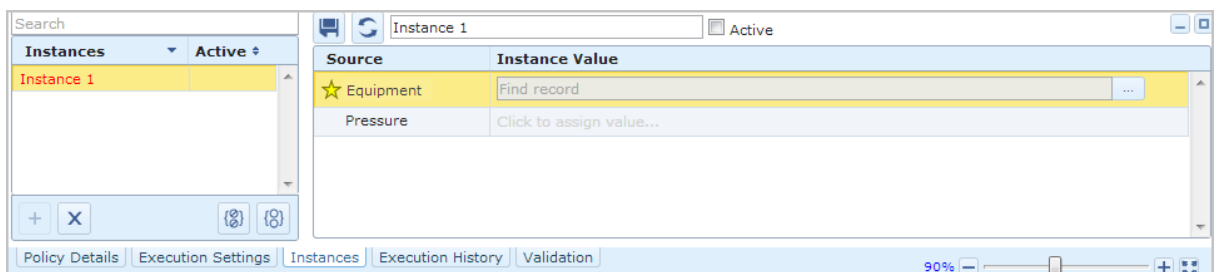
1. [Open the policy](#) associated with the policy instance for which you want to assign a record to a node.
2. On the [Instances tab](#), in the grid in the [Search area](#), select the row representing the policy instance for which you want to assign a record to a node. In the following image, *Instance 1* is selected in the grid in the Search area, and rows are displayed in the grid in the Details area for the *Equipment* and *Pressure* nodes.




Note: A yellow star appears next to the *Equipment* node name because the Equipment node has been [identified as the primary node](#).

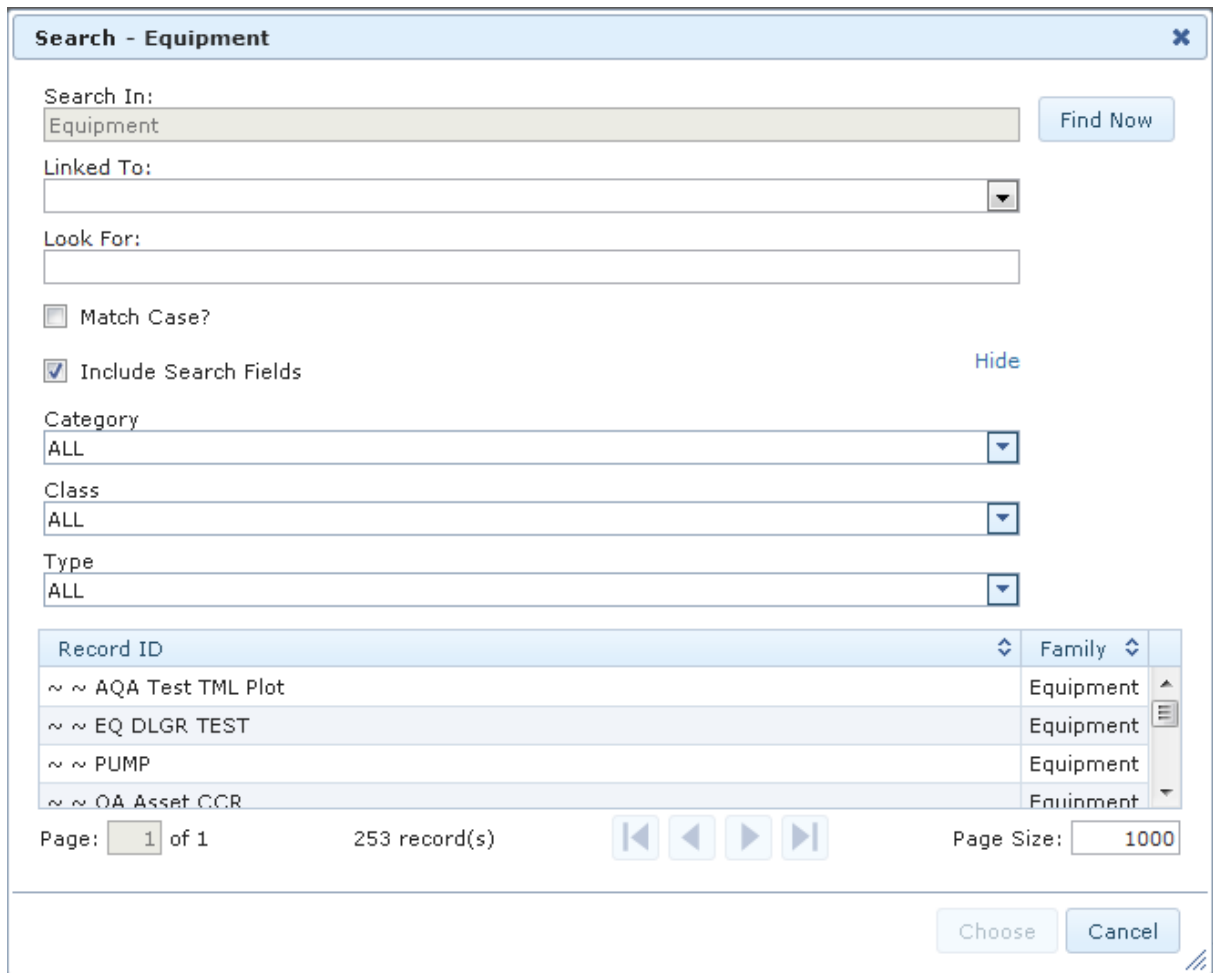
3. In the grid in the Details area, select the row representing the node to which you want to assign a record.

The row is highlighted in the grid, and the corresponding node is selected in the policy diagram. In the following image, the row associated with the *Equipment* node is selected.



- In the selected row, click the  button.

The **Search<Node>** window appears, where **<Node>** is the name of the policy node for which you accessed the window. The **Search In** text box contains the family caption of the family represented by the selected node. The search is run automatically, and a list of records that exist in that family are returned in the search results section.



Search - Equipment

Search In:

Linked To:

Look For:

Match Case?

Include Search Fields Hide

Category:

Class:

Type:

Record ID	Family
~ ~ AQA Test TML Plot	Equipment
~ ~ EQ DLGR TEST	Equipment
~ ~ PUMP	Equipment
~ ~ OA Asset CCR	Equipment

Page: of 1 253 record(s) Page Size:

If you have [specified the primary record in the policy](#), additional search criteria will be defined on this window by default that will return only records that are linked to the [primary record](#).

- If you want to limit the search results further, do so using the **Linked To** list and **Look For** text box.

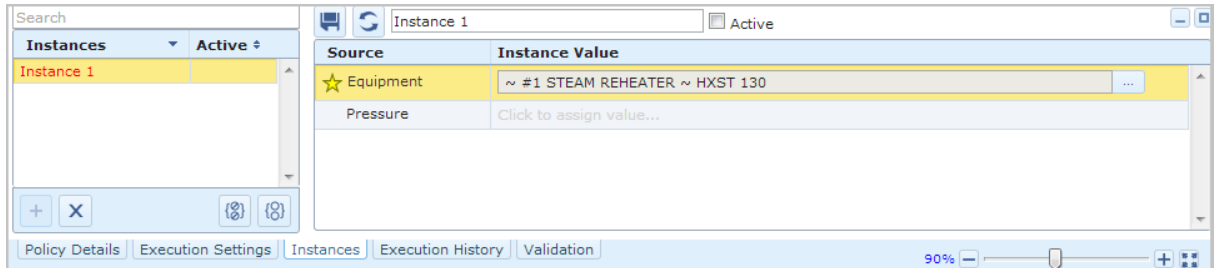
Note: Any text that you type in the **Look For** text box will apply to the families selected in *both* the **Search In** and **Linked To** fields.

- When you are finished specifying the search criteria, click the **Find Now** button.

The search results are filtered based upon your selections.

7. In the search results, select the row representing the record that you want to assign to the policy node, and click the **Choose** button.

The **Search <Node>** window closes, and the Record ID of the selected record appears in the **Instance Value** column in the grid in the Details area.



8. Repeat steps 3 through 7 until you have assigned a record to every row in the grid in the Details area.

Note: If you have specified a primary record in the policy, you will be able to type in the other cells in the **Instance Value** column, as opposed to using the **Search <Node>** window to select a record. When you do this, the type-ahead functionality displays a drop-down list of *up to* five records that are linked to the primary record *and* whose Record IDs contain the value that you typed in the cell. If you have not specified a primary record in the policy, you must use the **Search <Node>** window to select records to assign to nodes.

8. Click the  button.

The policy instance is saved.

Activating a Single Policy Instance

To activate a single policy instance:

1. [Open the policy](#) associated with the policy instance that you want to activate.
2. On the [Instances tab](#), in the grid in the [Search area](#), select the row representing the policy instance that you want to activate.

A grid appears in the [Details area](#), displaying all the nodes in the policy to which a specific record must be assigned (i.e., all Input nodes except for Query nodes and Constant nodes).

3. Above the grid in the Details area, select the **Active** check box.


The grid in the [Search area](#) is updated to display a check mark in the **Active** cell in the row representing this policy instance.

4. Above the grid in the Details area, click the  button.

The policy instance is activated. In addition, in the Policy Instance record, the value in the Active field is set automatically to *True*.

Activating All Policy Instances Associated with a Policy

To activate all policy instances associated with a policy:

1. [Open the policy](#) associated with the policy instances that you want to activate.
2. On the [Instances tab](#), below the grid in the [Search area](#), click the  button.

All policy instances that are associated with the policy are activated. In addition, in the Policy Instance records, the value in the Active field is set automatically to *True*.

Deactivating a Single Policy Instance


To deactivate a single policy instance:

1. [Open the policy](#) associated with the policy instance that you want to deactivate.
2. On the [Instances tab](#), in the grid in the [Search area](#), select the row representing the policy instance that you want to deactivate.

A grid appears in the [Details area](#), displaying all of the nodes in the policy to which a specific record must be assigned (i.e., all Input nodes except for Query nodes and Constant nodes).

3. Above the grid in the Details area, clear the **Active** check box.


The grid in the Search area is updated to remove the check mark from the **Active** cell in the row representing this policy instance.

4. Above the grid in the Details area, click the  button.

The policy instance is deactivated. In addition, in the Policy Instance record, the value in the Active field is set automatically to *False*.

Deactivating All Policy Instances Associated with a Policy

To deactivate all policy instances associated with a policy:

1. [Open the policy](#) associated with the policy instances that you want to deactivate.
2. On the [Instances tab](#), below the grid in the [Search area](#), click the  button.

All policy instances that are associated with the policy are deactivated. In addition, in the Policy Instance records, the value in the Active field is set automatically to *False*.

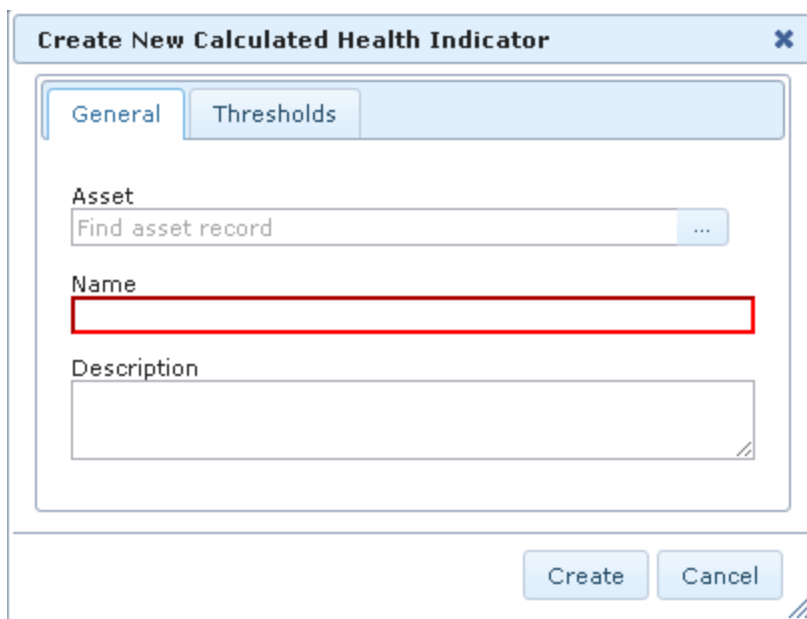
Creating a New Health Indicator Record from the Instances Tab

If a policy diagram contains a Health Indicator record, within a policy instance that is associated with that policy, you can assign an existing Health Indicator record to that node or create a new Health Indicator record to assign to that node. The following instructions explain how to create a new Health Indicator record that will be assigned to a Health Indicator node. You can then use the [Add Value to Health Indicator node](#) to create Health Indicator Value records and link them to the new Health Indicator record.

To create a new Health Indicator record that will be assigned to a Health Indicator node:

1. [Open the policy](#) associated with the policy instance for which you want to assign a new Health Indicator record to a Health Indicator node.
2. On the [Instances tab](#), in the grid in the [Details area](#), select the row representing the Health Indicator node to which you want to assign a new Health Indicator record.
3. In the **Instance Value** cell, click the [+](#) button.

The **Create New Calculated Health Indicator** window appears, displaying the **General** tab.



The screenshot shows a dialog box titled "Create New Calculated Health Indicator". It has two tabs: "General" and "Thresholds". The "General" tab is active. Inside the dialog, there are three input fields: "Asset" with a "Find asset record" button, "Name" (highlighted with a red border), and "Description". At the bottom right, there are "Create" and "Cancel" buttons.

4. In the **Asset** text box, click the [...](#) button.

The **Search** window appears.

The screenshot shows a 'Search' dialog box with the following elements:

- Search In:** A dropdown menu.
- Look For:** A text input field.
- Match Case?:** An unchecked checkbox.
- Page:** A field showing '1 of 0'.
- 0 record(s)**: Text indicating no results.
- Navigation:** Four buttons for navigating between pages (back, forward, first, last).
- Page Size:** A field showing '1000'.
- Buttons:** 'Find Now' button next to the 'Look For' field, and 'Choose' and 'Cancel' buttons at the bottom right.

5. In the **Search In** list, select the **Equipment** family.
6. In the **Look For** text box, type a keyword to limit your results to records containing that word within the Record ID. If you specify a keyword, your results will contain records that match your family criteria and contain the keyword that you specified. A keyword can contain any number of characters, any alphanumeric value, and any special character. A keyword is optional.
7. If you want the search to be case-sensitive, select the **Match Case** check box. Selecting the check box indicates that you want to return only the records that contain the keyword you specified using the *exact* case that you specified in the **Look For** text box.

-or-

If you do not want the search to be case-sensitive, do not select the **Match Case** check box.

Note: The **Match Case** option works only on Oracle schemas. When you perform a search using a SQL Server database, the search will be *case-insensitive* even when the **Match Case** check box is selected.

8. If additional search criteria appear below the **Match Case** check box:
 - If you want to refine the search results further to include only records with specific values in those fields, select the desired values in the available prompts.-or-
 - If you do not want to refine the search results further to include only records with specific values in those fields, clear the **IncludeSearch Fields** check box.

Note: If the underlying query prompts have been configured to contain the value *All* by default, if you do not want to refine the search results further, you can accept the default prompt values without clearing the **Include Search Fields** check box.

9. Click the **Find Now** button.

The search results appear.

10. In the search results, select the Equipment record to which you want to link the Health Indicator record, and click the **Choose** button.

The **Create New Calculated Health Indicator** window returns to focus, and the **Asset** text box is populated with the Record ID of the selected Equipment record.

The screenshot shows a dialog box titled "Create New Calculated Health Indicator". It has two tabs: "General" (selected) and "Thresholds". Under the "General" tab, there are three text input fields: "Asset", "Name", and "Description". The "Asset" field is populated with the text "04-S-M-51-13-501-Z90 ~ 0000000000001073749 ~ R ...". The "Name" field is empty and has a red border around it. The "Description" field is empty. At the bottom of the dialog, there are two buttons: "Create" and "Cancel".

11. In the **Name** field, type a name for the health indicator. This value is required.
12. In the **Description** field, type a brief description of the health indicator. This value is optional.
13. Click the **Thresholds** tab.

The screenshot shows a dialog box titled "Create New Calculated Health Indicator" with a close button (X) in the top right corner. Below the title bar are two tabs: "General" and "Thresholds". The "Thresholds" tab is active and contains seven text input fields arranged vertically. The fields are labeled as follows from top to bottom: "Upper Threshold 3", "Upper Threshold 2", "Upper Threshold 1", "Lower Threshold 1", "Lower Threshold 2", and "Lower Threshold 3". All input fields are currently empty. At the bottom right of the dialog box, there are two buttons: "Create" and "Cancel".

14. Type values in the threshold text boxes to indicate the thresholds against which values that are defined for this health indicator (that are stored in Health Indicator Value records) will be compared. For example, if Health Indicator Value records that are linked to this Health Indicator record will store a numeric value representing an overall health index on a scale of 1 to 10, where 1 represents critical health and 10 represents excellent health, you might define the thresholds like this:


This screenshot shows the same "Thresholds" tab as the previous image, but with some values entered into the input fields. From top to bottom, the fields contain the following values: "Upper Threshold 3" is empty, "Upper Threshold 2" is empty, "Upper Threshold 1" is empty, "Lower Threshold 1" contains the value "10", "Lower Threshold 2" contains the value "5", and "Lower Threshold 3" contains the value "1".

15. Click the **Create** button.

The new Health Indicator record is created, and the **Instances** tab is updated to show the Record ID of the new record in the **Instance Value** column in the grid in the Details area.

Saving Changes to a Policy Instance


To save changes to a policy instance:

- On the [Instances tab](#), above the grid in the [Details area](#), click the  button.

The changes that you have made to the policy instance are saved.

Deleting Policy Instances

To delete a policy instance:

1. [Open the policy](#) associated with the policy instance that you want to delete.
2. On the [Instances tab](#), in the grid in the [Search area](#), select the row representing the policy instance that you want to delete.
3. Below the grid, click the  button.

The policy instance is removed from the grid, and the Policy Instance record representing the policy instance is deleted.

About Validating the Policy Logic

While viewing a policy, you can trigger a validation process that will validate the policy logic automatically and display the results in a stand-alone window on the diagram canvas.

When the policy logic is validated, no actions will be taken (e.g., no Policy Recommendation records will be created). This prevents the policy from generating potentially invalid data while you are confirming that the policy logic is working as expected.

You can validate logic for one policy instance at a time. In order to validate the logic for a policy instance, values must be supplied to all Input nodes. The values can be provided in one of the following ways:

- [You can type the values manually.](#)
- [You can use values that are stored in the records that are included in a policy instance.](#)


Additionally, you can [copy the values in records belonging to the policy instance](#) if you want to:

- Use a combination of record values and values that you type manually for validation.
- or-
- Use record values as a starting point, but you plan to modify them before performing the validation.

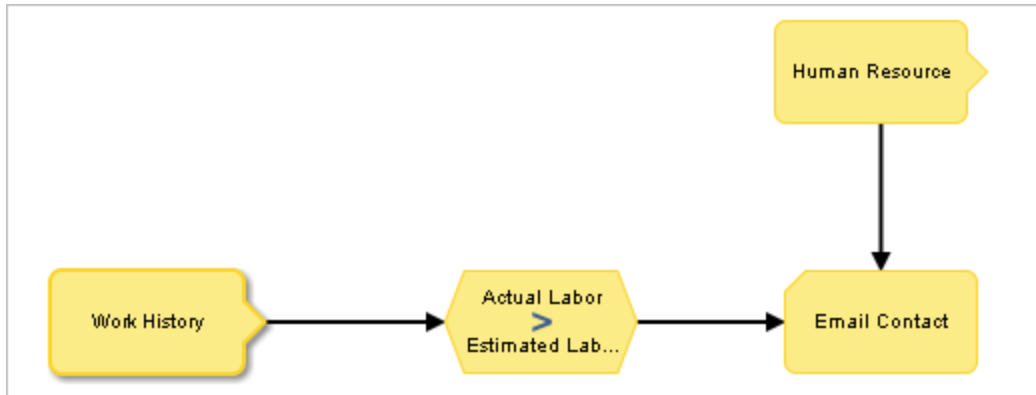
Validating the Logic Using Manual Values

The following instructions assume that the **Validation** tab is selected on the **Policy** page.

To validate the policy logic using manual values:

1. In the grid in the [Search area](#), select the row that contains the value *Ad Hoc* in the **Instances** column.
2. In the grid in the [Details area](#), in the **Values** column, type a value in each text box. These values will be used in the validation process.
3. Above the grid in the [Details area](#), click the  button.

The validation process begins. When the validation process is complete, the nodes are color-coded on the diagram canvas to indicate the outcome of the validation process. For example, in the following image, dark yellow is used to color nodes that passed the validation test without errors or warnings.



Validating the Logic Using Policy Instance Values

The following instructions assume that the **Validation** tab is selected on the **Policy** page and that the grid in the [Search area](#) contains instances for which values are displayed in the [Details area](#).

To validate the policy logic using values in the records belonging to a policy instance:

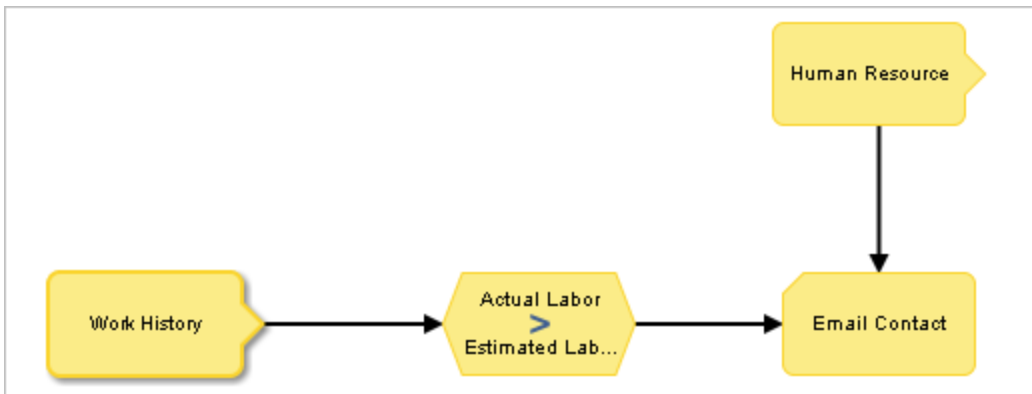
1. In the grid in the [Search area](#), select the row containing the policy instance whose logic you want to validate.

The grid in the [Details area](#) displays values from the records that are included in that policy instance. For example, in the following image, you can see that the value *20* is stored in the Actual Labor field in the Work History record included in the policy instance.

Input	Values
Work History	Actual Labor <input type="text" value="20"/> Estimated Labor <input type="text" value="15"/>
Human Resource	Email Address <input type="text" value="aduncan@meridium.com"/>

2. Above the grid in the [Details area](#), click the  button.

The validation process begins. When the validation process is complete, the nodes are color-coded on the diagram canvas to indicate the outcome of the validation process. For example, as shown in the following image, dark yellow is used to color nodes that passed the validation test without errors or warnings.



Copying Policy Instance Values for Validation

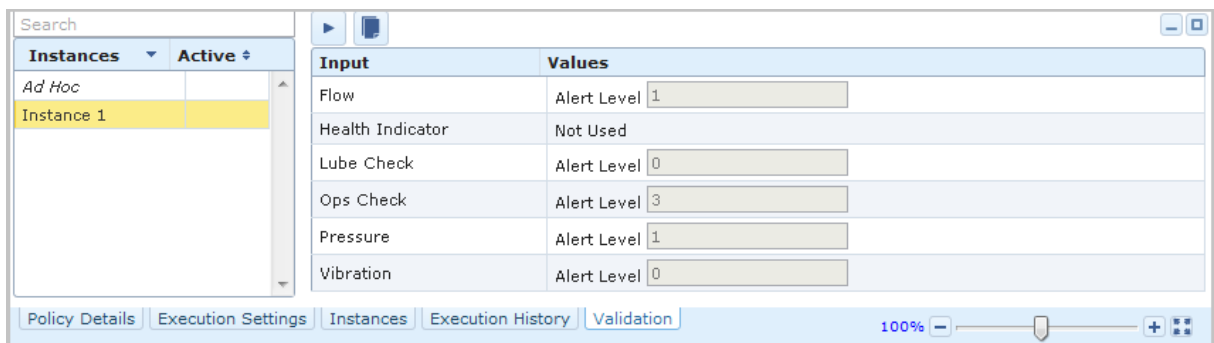
Via the **Validation** tab on the **Policy** page, you can copy the values in records belonging to the policy instance if you want to:

- Use a combination of those record values and values that you type manually for validation.
- or-
- Use those record values as a starting point, but modify them before performing the validation.

To copy policy instance values:

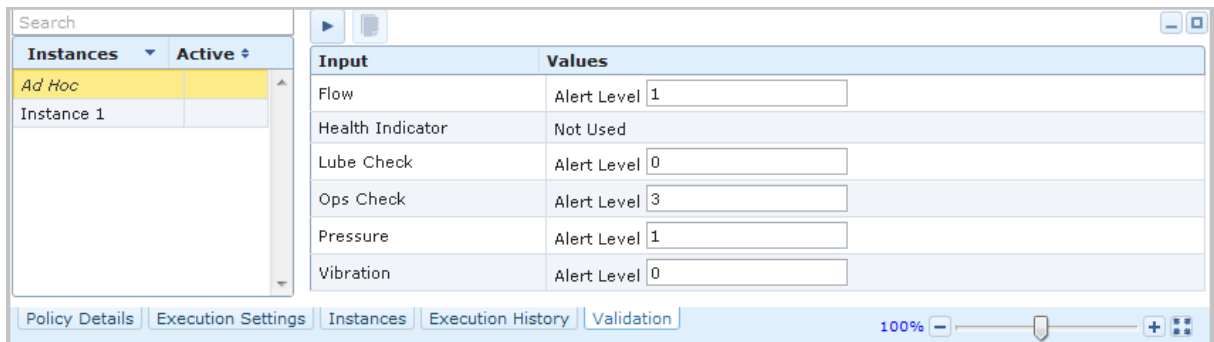
1. In the grid in the [Search area](#), select the row containing the policy instance whose values you want to copy.

For example, in the following image, the row associated with *Instance 1* is selected in the grid in the [Search area](#), and the input values associated with this policy instance are displayed in the grid in the [Details area](#).



2. Above the grid in the [Details area](#), click the  button.

The row associated with the *Ad Hoc* policy instance is selected in the grid in the [Search area](#), and the input values associated with *Instance 1* are used as the input values for the *Ad Hoc* policy instance.



At this point, the policy instance values have been copied. You can now [modify them manually](#) and then start the validation process.

V3.5.0 SP1 LP - Copyright © 1993-2012 Meridium, Inc. All rights reserved.

About Configuring Policies to be Executed

When a policy is executed, records or values that meet the conditions that are defined in the policy logic are used to trigger actions that are defined by the Action nodes in the policy diagram. Only [active policies](#) will be executed. Further, only the [active policy instances](#) that are associated with an active policy will be executed.

You can configure a policy to be executed:


- [When records included in the policy instances are updated.](#)
- [On a predefined schedule.](#)

Configuring Policies to Be Executed When Records Are Updated

The following instructions explain how to configure a policy to be executed when records belonging to a policy instance are updated. Note that:

- Only [active policies](#) will be executed.
- Only the [active policy instances](#) that are associated with an active policy will be executed.
- If the [Trigger check box](#) on the **Properties** window of a node is cleared, updates to the records that are associated with that particular node will *not* result in the execution of the policy even when the policy is configured to be executed when records belonging to a policy instance are updated.

To schedule a policy to be executed when records are updated:

1. [Open the policy](#) that you want to configure to be executed when records belonging to a policy instance are updated.
2. On the [Execution Settings tab](#), select the **Automatic Evaluation** check box.
3. Click the  button.

The policy is saved.

Configuring Policies to Be Executed on a Predefined Schedule

In order for a policy to be executed on a predefined schedule:

- An administrative user must [create a scheduled item](#) to define how often the system should check for policies that are due to be executed.
- You must define the policy execution schedule for the policy.

The following instructions explain how to define the policy execution schedule for an [active policy](#).

Note: The policy execution schedule should be *less* frequent than the schedule defined in the scheduled item. Otherwise, the policy will be executed only when the scheduled item runs. For example, if the scheduled item runs every day, the policy execution schedule should be less frequent than every day (e.g., every week).

To configure a policy to be executed on a predefined schedule:

1. [Open the policy](#) that you want to configure to be executed on a predefined schedule.
2. On the [Execution Settings tab](#), select the **Execute on a Schedule** check box.

The screenshot shows a configuration window titled "Execute on a Schedule" with a checked checkbox. Below the title is a large empty text input field with a blue ellipsis button to its right. Underneath the input field are two labels: "Next Date" on the left and "Last Date" on the right. At the bottom of the window, the text "Master Schedule Checks every 1 day(s) for policies due." is displayed.

3. Click the  button.

The **Schedule** window appears. By default, the value *One Time* is selected in the **Recurrence** list.

Schedule

Recurrence
One Time ▼

Starts
2012-10-11 11:12

Schedule Summary
Due one time effective 2012-10-11 11:12.

OK Cancel

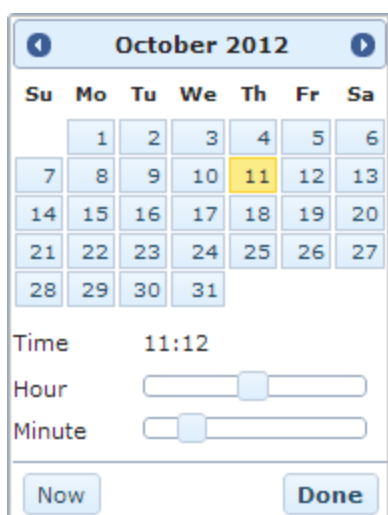
4. If you want the policy to be executed only one time, skip to step 6.
5. If you want the policy to be executed more than one time, to select the frequency by which the policy will be executed, in the **Recurrence** list, select the desired option, and complete the remaining options to define the schedule. The following table describes the schedule by which you can configure a policy to be executed.

Frequency	Options	Example	Default
Hourly	Every x hours.	Every five hours.	Every hour.
Daily	Every x days.	Every two days.	Every day.
	-or- Every weekday.	-or- Every weekday.	
Weekly	Every x weeks on <day(s)>.	Every two weeks on Monday and Friday.	Every week on Monday.
Monthly	Every x months on day n.	Every three months on day 15.	Every month on day 1.
	-or- Every x months on the n<day>.	-or- Every three months on the second Friday.	

Frequency	Options	Example	Default
Yearly	Every x years on <month><day> . -or- Every x years on n<day> of <month> .	Every year on July 1. -or- Every year on the first Friday of July.	Every year on January 1.

- If you want the policy to be executed starting on a date or time *other than* the current date and time, click inside the **Starts** text box.

A calendar appears.



The current date is highlighted on the calendar, and the current time is displayed below the calendar next to the **Time** label. The time format is determined by your selection in the **Time Zone** list on the **Execution Settings** tab.

- In the calendar, select the day on which the policy execution schedule should begin.
- Using the **Hour** slider, slide the square until the hour changes to the hour at which the policy execution schedule should begin.
- Using the **Minute** slider, slide the square until the minute changes to the minute at which the policy execution schedule should begin.
- Click the **Done** button to close the calendar.

The **Starts** text box is updated to reflect the selected date and time. In addition, a summary of the specified schedule appears below the **Starts** text box.

Schedule

Recurrence

Daily

Every 5 day(s)

Every weekday

Starts

2012-10-18 11:12

Schedule Summary

Due every 5 day(s) effective 2012-10-18 11:12.

OK Cancel

11. Click **OK**.

The **Execution Settings** tab on the **Policy** page returns to focus. A summary of the specified schedule and start time appears in the text box. In addition, the next date on which the policy will be executed appears below the **Next Date** label. The schedule that has been configured for the scheduled item appears below the **Master Schedule** label.


Execute on a Schedule

Due every 5 day(s) effective 2012-10-18 11:12.

Next Date: 2012-10-18 11:12

Last Date:

Master Schedule: No master schedule defined.

12. On the **Policy Actions** toolbar, click the  button.

The policy is saved.

About the Execution Results

The results of a policy execution are displayed on the [Execution History tab](#) on the [Policy](#) page. When you select a:

- Policy instance in the grid in the [Search area](#), the execution results for that instance are displayed in the Details area.
- Row in the grid in the [Details area](#), the nodes that are associated with that instance are [color-coded in the policy diagram to indicate the results of the execution](#).

The grid in the [Details area](#) displays the date and time that the policy was executed and a summary of the execution results. Summaries of the execution results include items such as warning messages, errors, returned values, and actions. For example, you might see the following summary information:

- **No Action Taken:** The policy was executed but no actions were triggered. For example, this might occur if the records belonging to the policy instance did not meet the conditions defined in the policy logic.

Start Time	Summary
2012-06-28 18:47	No action taken.

- **<Action Taken>:** The policy was executed and resulted in actions. The actions are listed in the **Summary** column of the grid (e.g., *Email sent to dhanson@meridium.com*).

Start Time	Summary
2012-06-20 18:01	Email sent to dhanson@meridium.com.

- **Errors Occurred. No Action Taken:** There was an error in the policy logic and no action was taken. For example, this might occur if an Email Contact node did not contain a value in the To Address field.

Start Time	Summary
2012-06-20 18:01	Errors occurred. No action taken.

You can [filter the results](#) to show only the executions that resulted in actions or only the executions that resulted in errors.

In the case of policy executions that result in a warning or error, the [Policy Execution Details window](#) is displayed automatically in the policy diagram. Additionally, in the policy diagram, you can select nodes that have been executed to [view additional details related to the execution](#).

Note: If changes have been made to the policy since it was last executed, you will not be able to view the results of the last execution process. The row in the grid in the

[Search area](#) will be disabled, and when you pause on the row, a message will appear stating that changes have been made and the execution data cannot be displayed.

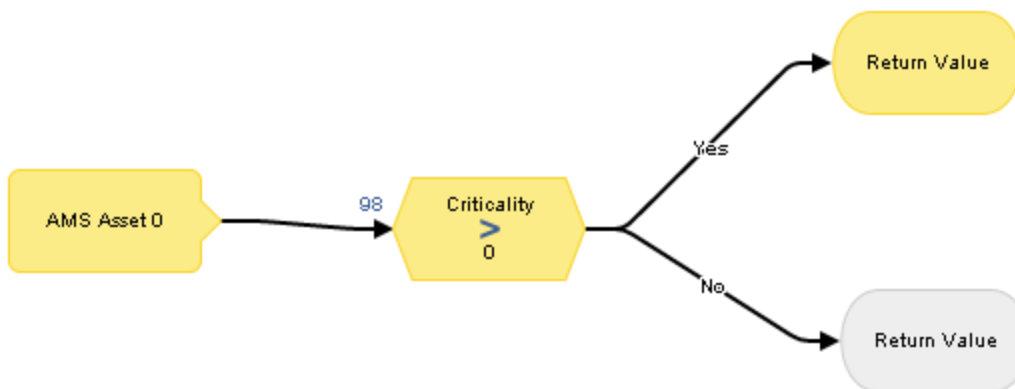
Interpreting Node Color-Coding in the Policy Diagram

After a policy has been executed, when you select a row in the Details area on the **Execution History** tab, the nodes in the diagram will be color-coded as follows to indicate the results of the execution process:

- **Dark Yellow:** Indicates that the node was executed successfully, (i.e., the source value was valid and produced the intended result).
- **Gray:** Indicates that the node was not executed.
- **Pink:** Indicates that the node was not executed successfully because the source value caused an error to occur.
- **Pale yellow:** Indicates that the source node was not executed successfully because the source value caused a warning to occur.

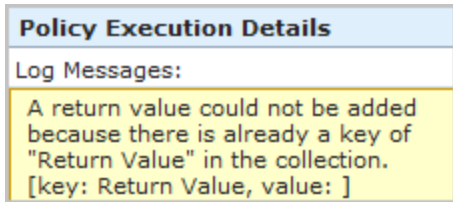
For example, the following image illustrates that:

- The *AMS Asset*, *Condition*, and top *Return Value* nodes were executed successfully.
- The bottom *Return Value* node was not executed.

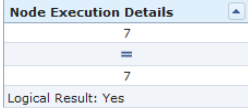


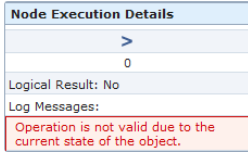
Aspects of the Execution Details Windows

When a policy execution results in warning or errors, the **Policy Execution Details** window appears automatically on the diagram canvas. The **Policy Execution Details** window displays the specific error or warning message. The following image shows an example of the **Policy Execution Details** window.



Likewise, on the diagram canvas, when you select a node that has been executed, the **Node Execution Details** window appears on the diagram canvas. The following table provides details on the information that is displayed in the **Node Execution Details** window depending upon the outcome of the execution.

If the:	...The Node Execution Details window displays:	For example:
Node was executed successfully.	The output of the node (e.g., an input value, logical result, or action).	<p>The following image shows an example of the Node Execution Details window for a Condition node that was executed successfully with a logical result of Yes (e.g., seven equals seven).</p>  <p>Note that the same node could have also been executed successfully with a logical result of No (e.g., eight is not equal to seven).</p>

If the:	...The Node Execution Details window displays:	For example:
Node execution resulted in an error.	A log message that is colored red.	<p>For example, the following image shows an example of the Node Execution Details window for a Condition node that was executed and resulted in an error (i.e., Null cannot be compared to 0).</p>  <p>The screenshot shows a window titled 'Node Execution Details'. It contains a right-pointing arrow, the value '0', and the text 'Logical Result: No'. Below this, under 'Log Messages:', there is a red-bordered box containing the text: 'Operation is not valid due to the current state of the object.'</p>
Node execution resulted in a warning.	A log message that is colored yellow.	N/A

Dashboards Folder

The following table lists the dashboards that are stored in the Catalog location \\Public\Meridium\Modules\Policy Manager\Dashboards. The dashboard name is listed in the table. The dashboard name and caption are the same.

Dashboard	Behavior and Usage
Policy Manager Start Page	Displays the baseline Policy Manager dashboard on the Dashboard page, which serves as the start page for the Policy Manager module.

Queries Folder

The following table lists the queries that are stored in the Catalog location \\Public\Meridium\Modules\Policy Manager\Queries. The queries are listed in the table by name. The query name and caption are the same unless otherwise noted.

Query	Behavior and Usage
Email Address Query	Returns values in the First Name, Last Name, and Email Address fields in Human Resource records whose Email Address field contains a value. This query is used to display results in the grid on the Choose User window.
Events Query	<p>Returns information about the Severity, Name, Description, Type, Start Time, End Time, and Duration (in seconds) of Policy Events. This query can be used with the Query node in the policy diagram to bring in policy events related to pieces of equipment or functional locations.</p> <p>The Events Query includes an <i>Asset Key</i> prompt, which can be passed to the Query node via the Entity Key system field of a family-based predecessor node.</p> <p>By default, the results of the Events Query are sorted in ascending order by start time.</p> <p>This query works with an SQL database. See Events Query (Oracle) for use with an Oracle database.</p>
Events Query (Oracle)	Returns the same values as the Events query. This query, however, works in an Oracle database.
Policies Address Query	Returns values in the Policy Name, Policy Description, and Active fields of a Policy record. This query is used to display results on the Policies Search page .

Dashboard Queries Folder

The following table lists the queries that are stored in the Catalog location \\Public\Meridium\Modules\Policy Manager\Queries\Dashboard Queries. The queries are listed in the table by name. The query name and caption are the same unless otherwise noted.

Query	Behavior and Usage
My Policy Recommendations	<p>Returns all Policy Recommendation records that are:</p> <ul style="list-style-type: none"> Assigned to the Security User who is currently logged in. -and- <i>Not</i> in a Closed state. <p>The results of this query appear in the My Policy Recommendations section on the Policy Manager Start Page.</p>
My Recent Policies	<p>Returns Policy records that were last updated by the Security User who is currently logged in within the last 30 days. The results of this query are displayed in the My Recent Policies section on the Policy Manager Start Page.</p>
Policy Executions	<p>Returns Policy records that have been executed within the last 30 days. The results of this query are displayed in the Policy Executions section on the Policy Manager Start Page.</p>
Policy Recommendations - 30 days past due	<p>Returns all Policy Recommendation records:</p> <ul style="list-style-type: none"> Whose Target Completion Date field contains a value that corresponds with a date that is at least 30 days prior to today's date. -and- That are <i>not</i> in a Closed state. <p>The results of this query are displayed in the Policy Recommendations section on the Policy Manager Start Page.</p>
Policy Results	<p>Returns all Policy records that have been executed within the last 30 days. The results of this query are displayed in the Policy Results section on the Policy Manager Start Page.</p>

Policy Manager Families Delivered with Read-Only Properties

To ensure that the Policy Manager module functions as intended, the following Policy Manager families are delivered with read-only properties:

- [Policy](#)
- [Policy Event](#)
- [Policy Instance](#)

Policy

Policy records store basic information about policies. The following table provides an alphabetical list and description of the fields that exist in the Policy family and that are displayed on the baseline Policy datasheet (unless otherwise noted). The information in the table reflects the baseline state and behavior of these fields.

Note:The values in Policy records cannot be modified in Record Manager. You must view the policy in the Policy Manager module in order to change values in the record.

Field	Data Type	Description	Behavior and Usage
Active	Logical	Indicates whether the policy is activate or inactive.	<p>Reflects the selection in the Active check box on the Execution Settings tab on the Policy page. When the check box is:</p> <ul style="list-style-type: none"> • <i>Selected</i>, the value in this field is set to <i>True</i>, meaning that the policy is active and can be executed. • <i>Cleared</i>, the value in this field is set to <i>False</i>, meaning that the policy is inactive and cannot be executed.
Automatic Evaluation	Logical	Indicates that the policy logic will be executed automatically when records belonging to the policy instance are updated.	<p>Reflects the selection in the Automatic Evaluation check box on the Execution Settings tab on the Policy page. When the check box is:</p> <ul style="list-style-type: none"> • <i>Selected</i>, the value in this field is set to <i>True</i>, meaning that the policy logic will be executed every time a record is modified. • <i>Cleared</i>, the value in this field is set to <i>False</i>, meaning that the policy logic will not be executed every time a record is modified.

Field	Data Type	Description	Behavior and Usage
Description	Text	A brief summary of the policy.	Contains the value that appears in the Description text box on the Policy Details tab on the Policy page.
Last Scheduled Execution	Date	The most recent date on which the policy was executed.	This field is populated in one of the following ways, depending upon whether or not a schedule has been defined for the policy and whether or not the policy has been executed. If a schedule: <ul style="list-style-type: none"> • Is not defined for the policy or the policy has not yet been executed, this field is empty. • Has been defined for the policy and the policy has been executed at least once, this field is populated automatically with the Last Date value that appears on the Execution Settings tab on the Policy page.
Model	Binary	Stores code defining the logic that is represented by the policy diagram.	This field is disabled and does not appear on the Policy datasheet.
Name	Character	The name of the policy.	Contains the value that appears in the Name text box on the Policy Details tab on the Policy page. This field is required, and the value must be unique.

Field	Data Type	Description	Behavior and Usage
Next Scheduled Execution	Date	The next date that the policy is scheduled to be executed.	<p>This field is populated in one of the following ways, depending upon whether or not a schedule has been defined for the policy. If a schedule:</p> <ul style="list-style-type: none"> • Is not defined for the policy, this field is empty. • Has been defined for the policy, this field is populated automatically with the Next Date value that appears on the Execution Settings tab on the Policy page.
Owner	Character	The owner of the policy.	<p>Contains the Policy Owner value that appears on the Policy Details tab on the Policy page. Only the owner of the policy can save changes to the policy.</p> <p>This field is required. This field is configured to track revision history by default. You can look at the revision history for this field to determine the ownership history for a particular policy.</p>
Schedule	Character	A summary of the policy schedule.	<p>This field is populated in one of the following ways, depending upon whether or not a schedule has been defined for the policy. If a schedule:</p> <ul style="list-style-type: none"> • Is not defined for the policy, this field is empty. • Has been defined for the policy, this field contains the value in the Execute on a Schedule text box on the Execution Settings tab on the Policy page.

Field	Data Type	Description	Behavior and Usage
Scheduled	Logical	Indicates whether or not the policy has a defined execution schedule.	<p>This field is populated in one of the following ways, depending upon whether or not a schedule has been defined for the policy. If a schedule:</p> <ul style="list-style-type: none">• Is not defined for the policy, this field is set to <i>False</i>.• Has been defined for the policy, the value is set to <i>True</i>.
Timezone	Character	The time zone associated with the policy.	<p>This field contains the value (<i>UTC Coordinated Universal Time</i>) by default. You can modify this value by selecting a different value in the Time Zone list on the Executions Settings tab on the Policy page.</p>

Policy Event

Policy Event records store information about [events that are associated with Equipment or Functional Location records that are monitored by a policy](#). The following table provides an alphabetical list and description of the fields that exist in the Policy Event family and that are displayed on the baseline Policy datasheet (unless otherwise noted). The information in the table reflects the baseline state and behavior of these fields. This list is *not* comprehensive.

Field	Data Type	Description	Behavior and Usage
Close Description	Text	A description of why the event was closed.	Populated with the value that you specify on the Properties window for a Create Event or Close Event node.
Description	Text	A description of the event.	Populated with the value that you specify on the Properties window for a Create Event node. This value will appear as a hyperlink in the Summary column in the grid on the Execution History tab that you can click to access the Events tab on the Asset Health Manager page.
End Time	Date	The date on which the event ended. This field is used only when the Has Duration field is set to <i>True</i> .	Populated with the value that you specify on the Properties window for a Create Event or Close Event node.

Field	Data Type	Description	Behavior and Usage
Event Type	Character	<p>The type of event. This field contains one of the following values:</p> <ul style="list-style-type: none"> • Generic: Indicates that the event is not associated with a specific type of event. • Excursion: Indicates that the event is associated with an operation that is outside of an established operating window. 	Populated with the value that you specify on the Properties window for a Create Event node.
Has Duration	Logical	Indicates whether or not there is an end time associated with the event.	Populated with the value that you specify on the Properties window for a Create Event node.
Name	Character	The name of the event.	Populated with the value that you specify in the Event Name section on the Properties window for a Create Event node.
Policy Instance GUID	Binary	Contains code that is used internally by the Meridium APM system.	This field is disabled.

Field	Data Type	Description	Behavior and Usage
Severity	Character	<p>The severity of the event.</p> <p>This field contains one of the following values:</p> <ul style="list-style-type: none"> • Information: Indicates that the event exists simply to provide information about activity conducted for the associated piece of equipment or location. • Warning: Indicates that the event exists to communicate some sort of low-risk warning about the associated piece of equipment or location. • Alert: Indicates that the event exists to communicate some sort high-risk warning about the associated piece of equipment or location. 	Populated with the value that you specify on the Properties window for a Create Event node.
Start Time	Date	The date on which the event started.	Populated with the value that you specify on the Properties window for a Create Event node.
Time Line Reset	Logical	This field is not currently used.	N/A

Policy Instance

Policy Instance records store information about policy instances that are associated with a policy. The following table provides an alphabetical list and description of the fields that exist in the Policy Instance family. The information in the table reflects the baseline state and behavior of these fields.

Note: The values in Policy Instance records cannot be modified in Record Manager. You must update the instance in the Policy Manager module in order to change values in the record.

Field	Data Type	Description	Behavior and Usage	Datasheet
Active	Logical	Indicates whether the instance is activate or inactive.	<p>Reflects the selection in the Active check box on the Instances tab on the Policy page.</p> <ul style="list-style-type: none"> When the check box is selected, the value is set to <i>True</i>, meaning that the instance is active and can be executed. When the check box is cleared, the value is set to <i>False</i>, meaning that the instance is inactive and cannot be executed. 	Policy Instance
ID	Character	The name of the instance.	<p>Contains the value that you type in the Instance text box on the Instances tab on the Policy page. The default value for this field is <i>Instance <#></i>, where <#> is a number representing the number of policy instances that are already associated with the current policy.</p>	Policy Instance

Field	Data Type	Description	Behavior and Usage	Datasheet
Mapping	Binary	Contains code that maps values from records belonging to the policy instance to policy diagram nodes.	This field is disabled.	None
Policy GUID	Binary	Contains code used internally by the Meridium APM system.	This field is disabled.	None

Policy Recommendation

Policy Recommendation records store basic information about [recommendations that have been created as a result of a policy](#). The following table provides an alphabetical list and description of some of the fields that exist in the Policy Recommendation family and appear on the Policy Recommendation datasheet. The information in the table reflects the baseline state and behavior of these fields. This list is not comprehensive.

Field	Data Type	Description	Behavior and Usage
Create Work Request?	Logical	Specifies whether or not a work request for the EAM system that you have configured in Meridium APM will be created from the Policy Recommendation record.	Populated with the value that you specify on the Properties window for the Create Recommendation node.
Completed Date	Date	The date on which the recommended action was completed.	You can use the Calendar feature to manually select the date on which the recommended action was completed.
Completion Comments	Text	Details about the completed recommendation.	You can type a value manually.
Equipment ID	Character	The Record ID of the Equipment record to which the Policy Recommendation record is linked.	Populated with the value that you specify on the Properties window for the Create Recommendation node.

Field	Data Type	Description	Behavior and Usage
Functional Location ID	Character	The Record ID of the Functional Location record to which the Policy Recommendation record is linked. -or- The Record ID of the Functional Location record that is linked to the Equipment record to which the Policy Recommendation record is linked.	Populated with the value that you specify on the Properties window for the Create Recommendation node.
Originating Reference	Character	The Reference ID of the event or any other entity that originated the Policy Recommendation record.	Populated with the value that you specify on the Properties window for the Create Recommendation node.
Recommendation Basis	Character	The policy that created the recommendation.	Populated automatically with the name of the Policy record to which the Policy Recommendation record is linked.
Recommendation Description	Text	Information about the policy logic that caused the Policy Recommendation record to be created.	Populated automatically.
Recommendation Headline	Character	A short description of the recommended action.	Populated with the value that you specify on the Properties window for the Create Recommendation node.

Field	Data Type	Description	Behavior and Usage
Recommendation ID	Character	A unique value that identifies the Policy Recommendation record.	Populated automatically and disabled.
Recommendation Priority	Character	The value used to rank importance of the Policy Recommendation record. On the datasheet, this field contains a list of values that you have specified via field-level rules.	Populated by the value you specify on the Properties window for the Create Recommendation node.
Recommendation Type	Character	The type of recommendation record.	Populated automatically with the value <i>Policy (PCY)</i> .
Target Completion Date	Date	The date by which the recommended action should be completed.	Populated with the value that you specify on the Properties window for the Create Recommendation node.
Work Request Equipment	Character	The ID of the SAP, Oracle EBS eAM, or Maximo Equipment that is associated with the work request that was created from this Policy Recommendation record.	Disabled and populated automatically after the work request is created.
Work Request Functional Location	Character	The ID of the SAP, Oracle EBS eAM, or Maximo Functional Location that is associated with the work request that was created from this Policy Recommendation record.	Disabled and populated automatically after the work request is created.

Field	Data Type	Description	Behavior and Usage
Work Request Reference	Character	The ID of the SAP, Oracle EBS eAM, or Maximo work request that was created from this Policy Recommendation record.	Disabled and populated automatically after the work request is created.

Parameters for the Policy Manager URL

There is one root URL associated with Policy Manager: **meridium://Policies**. A link constructed from the path alone will open the **Policy Manager Start Page**. This URL accepts the parameters that are described in the following table.

Parameter Name	Description	Accepted Value (s)	Notes
EntityKey	Specifies the Policy record that you want to open on the Policy page.	0 -or- The Entity Key of the desired Policy record.	<ul style="list-style-type: none"> • 0: Displays a new policy on the Policy page. • <EntityKey>: Displays the Policy record with the specified Entity Key on the Policy page.
InstanceKey	Specifies the policy instance that you want to open.	The Entity Key of the desired Policy Instance record.	None
Page	Specifies the page that you want to view.	Search	Displays the Policies Search page .

Examples of Policy Manager URLs

- **meridium://Policies**
Displays the **Policy Manager Start Page**.
- **meridium://Policies?Page=Search**
Displays the **Policies Search** page.
- **meridium://Policies?EntityKey=0**
Displays a new policy on the **Policy** page.
- **meridium://Policies?EntityKey=3488054**
Displays the **Policy** page for the Policy record with the Entity Key 3488054.
- **meridium://Policies?EntityKey=3488054&InstanceKey=5798423**
Displays the **Policy** page for the Policy record with the Entity Key 3488054. The **Instances** tab is selected automatically, and the Policy Instance record identified by the Entity Key 5798423 will be selected in the grid.

Policy

A strategic plan that defines actions that should be taken when certain conditions exist. A *policy* consists of a [Policy record](#) and the policy diagram defining the policy logic.

Policy Diagram

A collection of [nodes](#) and connections that are added to the diagram canvas to define the policy logic.

Policy Instance

A collection of records whose values you want to monitor (e.g., Equipment 101). A *policy instance* consists of a Policy Instance record and the records that belong to the policy instance, where each policy instance includes one record for *each* input node in the policy diagram.

For example, assume that a policy diagram contains two Entity nodes: one for the *Equipment* family and one for the *Work History* family. In this case, a policy instance that is associated with this policy will include two records: *Tank 101* (Equipment record) and *Repair - Tank 101* (Work History record).

Primary Record

The record by which search results are filtered when you assign a record to a node. Specifically, when you search for the record that you want to assign to a node, the search results will return only records that are linked to the record that is designated as the [primary record](#).

For example, consider a policy that contains two Entity nodes: one for the *Equipment* family and one for the *Work History* family. You could assign an Equipment record to the Equipment node and specify that the Equipment record is the primary record. After doing so, when you search for a Work History record to assign to the Work History node, the search results will return only Work History records that are linked to that Equipment record.

Primary Node

The node to which [the primary record has been assigned](#).