Quest

Foglight® for Integrations 7.3.0

User and Reference Guide

© 2024 Quest Software Inc.

ALL RIGHTS RESERVED.

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Quest Software Inc.

The information in this document is provided in connection with Quest Software products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Quest Software products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, QUEST SOFTWARE ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL QUEST SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF QUEST SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Quest Software makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Quest Software does not make any commitment to update the information contained in this document.

If you have any questions regarding your potential use of this material, contact:

Quest Software Inc.
Attn: LEGAL Dept.
4 Polaris Way
Aliso Viejo, CA 92656

Refer to our website (https://www.quest.com) for regional and international office information.

Patents

Quest Software is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at https://www.quest.com/legal.

Trademarks

Quest, the Quest logo, and Where next meets now are trademarks and registered trademarks of Quest Software Inc. For a complete list of Quest marks, visit https://www.quest.com/legal/trademark-information.aspx. "Apache HTTP Server", Apache, "Apache Tomcat" and "Tomcat" are trademarks of the Apache Software Foundation. Google is a registered trademark of Google Inc. Android, Chrome, Google Play, and Nexus are trademarks of Google Inc. Red Hat, JBoss, the JBoss logo, and Red Hat Enterprise Linux are registered trademarks of Red Hat, Inc. in the U.S. and other countries. CentOS is a trademark of Red Hat, Inc. in the U.S. and other countries. Fedora and the Infinity design logo are trademarks of Red Hat, Inc. Microsoft, .NET, Active Directory, Internet Explorer, Hyper-V, Office 365, SharePoint, Silverlight, SQL Server, Visual Basic, Windows, Windows Vista and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. AIX, IBM, PowerPC, PowerVM, and WebSphere are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Java, Oracle, Oracle Solaris, PeopleSoft, Siebel, Sun, WebLogic, and ZFS are trademarks or registered trademarks of Oracle and/or its affiliates in the United States and other countries. SPARC is a registered trademark of SPARC International, Inc. in the United States and other countries. Products bearing the SPARC trademarks are based on an architecture developed by Oracle Corporation. OpenLDAP is a registered trademark of the OpenLDAP Foundation. HP is a registered trademark that belongs to Hewlett-Packard Development Company, L.P. Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both. MySQL is a registered trademark of MySQL AB in the United States, the European Union and other countries. Novell and eDirectory are registered trademarks of Novell, Inc., in the United States and other countries. VMware, ESX, ESXi, vSphere, vCenter, vMotion, and vCloud Director are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions. Sybase is a registered trademark of Sybase, Inc. The X Window System and UNIX are registered trademarks of The Open Group. Mozilla and Firefox are registered trademarks of the Mozilla Foundation. "Eclipse", "Eclipse Foundation Member", "EclipseCon", "Eclipse Summit", "Built on Eclipse", "Eclipse Ready" "Eclipse Incubation", and "Eclipse Proposals" are trademarks of Eclipse Foundation, Inc. IOS is a registered trademark or trademark of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries. Apple, iPad, iPhone, Mac OS, Safari, Swift, and Xcode are trademarks of Apple Inc., registered in the U.S. and other countries. Ubuntu is a registered trademark of Canonical Ltd. Symantec and Veritas are trademarks or registered trademarks of Symantec Corporation or its affiliates in the U.S. and other countries. OpenSUSE, SUSE, and YAST are registered trademarks of SUSE LCC in the United States and other countries. Citrix, AppFlow, NetScaler, XenApp, and XenDesktop are trademarks of Citrix Systems, Inc. and/or one or more of its subsidiaries, and may be registered in the United States Patent and Trademark Office and in other countries. AlertSite and DéjàClick are either trademarks or registered trademarks of Boca Internet Technologies, Inc. Samsung, Galaxy S, and Galaxy Note are registered trademarks of Samsung Electronics America, Inc. and/or its related entities. MOTOROLA is a registered trademarks of Motorola Trademark Holdings, LLC. The Trademark BlackBerry Bold is owned by Research In Motion Limited and is registered in the United States and may be pending or registered in other countries. Quest is not endorsed, sponsored, affiliated with or otherwise authorized by Research In Motion Limited. Ixia and the Ixia four-petal logo are registered trademarks or trademarks of Ixia. Opera, Opera Mini, and the O logo are trademarks of Opera Software ASA. Tevron, the Tevron logo, and CitraTest are registered trademarks of Tevron, LLC. PostgreSQL is a registered trademark of the PostgreSQL Global Development Group. MariaDB is a trademark or registered trademark of MariaDB Corporation Ab in the European Union and United States of America and/or other countries. Vormetric is a registered trademark of Vormetric, Inc. Intel, Itanium, Pentium, and Xeon are trademarks of Intel Corporation in the U.S. and/or other countries. Debian is a registered trademark of Software in the Public Interest, Inc. OpenStack is a trademark of the OpenStack Foundation. Amazon Web Services, the "Powered by Amazon Web Services" logo, and "Amazon RDS" are trademarks of Amazon.com, Inc. or its affiliates in the United States and/or other countries. Infobright, Infobright Community Edition and Infobright Enterprise Edition are trademarks of Infobright Inc. POLYCOM®, RealPresence® Collaboration Server, and RMX® are registered trademarks of Polycom, Inc. All other trademarks and registered trademarks are property of

their respective owners.

Legend

WARNING: A WARNING icon indicates a potential for property damage, personal injury, or death.

CAUTION: A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.

IMPORTANT NOTE, **NOTE**, **TIP**, **MOBILE**, or **VIDEO:** An information icon indicates supporting information.

Contents

Using Foglight for Integrations 6
Understanding the Workflow 6
Alarms Workflow 6
Service-Oriented Workflow 7
SNMP Trap Receiver Workflow 7
External Acknowledgement and Clearing of Alarms Workflow 8
Configuring Foglight for Integrations 8
Working in the Alarm Properties Dashboard 8
Adding Configuration Item Properties 9
Editing Alarm and Configuration Item Properties 10
Integration Samples 10
Receiving Alarms Samples 10
Forwarding Alarms Samples 11
Obtaining Integration Samples 12
Receiving Third-Party Alarms in Foglight 12
Preparing the Data 12
Viewing Third-Party Alarms 14
Working with Configuration Items 19
Using the CI Model in Service Builder 22
Forwarding Alarms to Third-Party Systems 23
Forwarding Alarms 23
Creating an XMLDump Script 27
Receiving SNMP Traps 27
Loading MIB Definitions 28
Configuring Alarms from SNMP Trap Definitions 29
Grouping SNMP Traps 31
Configuring Agent Properties for the SNMP Trap Agent 32
Troubleshooting 32
Receiving Acknowledged and Cleared Commands 33
Viewing Alarms 34
Deploying and Configuring QMX Agents 34
Deploying QMX Agents 34
Configuring QMX Agent Properties 35
Configuring QMX 36
Configuring ServiceNow Integration 37
Requirements 37
Configuring ServiceNow integration 37

Reference 41
Views 41
CI List View 41
Alarm Properties View 42
CI Properties View 44
Configuration View 45

SNMP Trap Groups View 47
Rules 47
Alarm Integration Rule 48
Clear Integration Alarm Rule 48
Incident Integration Rule 48

About Us 50
Technical support resources 50

1

Using Foglight for Integrations

This User and Reference Guide provides instructions, and conceptual information about how to receive alarms from third-party systems in to Foglight. Learn how to receive Simple Network Management Protocol (SNMP) Traps into Foglight as alarms, and how to forward alarms from Foglight to third-party systems. Information on how external systems can format XML documents and feed them into Foglight to perform acknowledge and clear functions on alarms is provided.

In addition, this guide provides information about the dashboards, rules, and views that are available for your monitored Foglight for Integrations infrastructure.

This guide is intended for an expert user.

Understanding the Workflow

Foglight for Integrations enables you to receive alarms from third-party systems in to Foglight, and to forward alarms from Foglight to third-party systems. In addition, Foglight for Integrations contains a Simple Network Management Protocol (SNMP) Trap Receiver that can convert traps into alarms based on their Management Information Base (MIB) definitions and associated configurations. The following topics outline the different workflows for the Foglight for Integrations.

Alarms Workflow

Alarm workflows involve examining a set of alarms in real time and acting on those alarms. Operators who use the alarm workflows are interested in all alarms on some set of items. They watch the trend of those alarms over time and the most recent alarms. When alarms appear, the Operator takes immediate action by either acknowledging the alarm, clearing the alarm (to close an alarm that is no longer active), and forwarding the problems for resolution. The Alarms dashboard is best suited for system-wide alarm-oriented workflows.

User-defined properties can be supplied on third-party alarms and Configuration Items. These properties can be used to provide additional details about the alarm or Configuration Item.

A transformer concept is used to send alarms to Foglight. It transforms the data from the third-party systems format to the XML format that Foglight requires. During that transformation, additional properties can be formatted by using key-value pairs.

The key is the property name and the value is the value of the property. Before sending in a new property, it must be defined in Foglight. This is accomplished using the CI Property dashboard or the Alarm Property dashboard.

After the properties are fed into Foglight, the Configuration Items are displayed in a new model called the *CIModel*. You can search for Configuration Items either by using a Data browser interface or by viewing the list in the Integration dashboard.

These Configuration Items can then be included in services using traditional functionality already in Service Builder dashboard. Alarms received from the third-party systems would then impact the appropriate services.

Service-Oriented Workflow

A service is a collection of objects that you want to monitor. Users create most services on the basis of what they are responsible for, and are typically organized around what an Operator needs to monitor.

Foglight provides a feature called a Service Builder which allows the user to group one or more components. It provides the functions needed to create a service and edit existing services. When you create a service, a corresponding service level is automatically created.

Services are used as inputs to many other dashboards (Hosts Table, Agents) besides the Services dashboards, as well as to reports. Defining a good set of services can make other dashboards more useful and easier to understand.

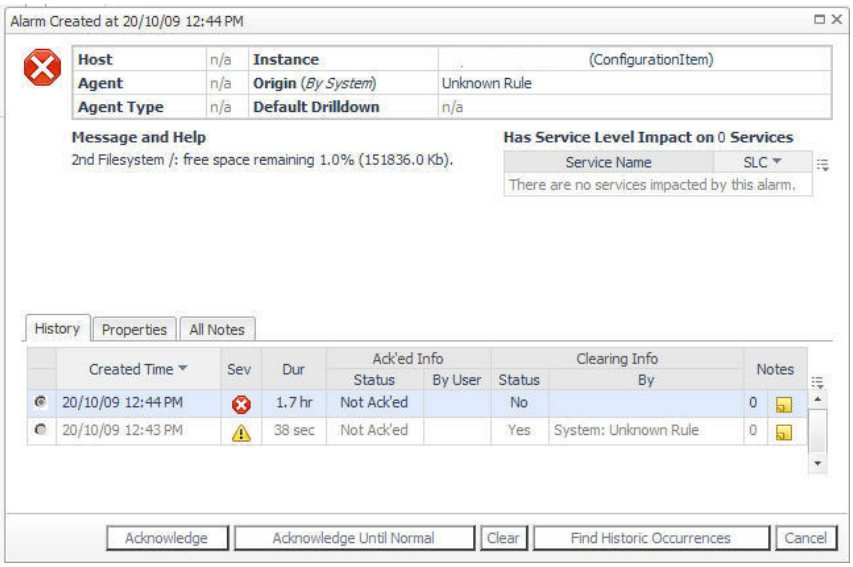
For more information, see Using the CI Model in Service Builder on page 22.

SNMP Trap Receiver Workflow

Foglight for Integrations includes default MIBs containing trap definitions that can be used to format Simple Network Management Protocol (SNMP) Traps that Foglight receives as third-party Alarms. By receiving SNMP traps, you are able to capture information regarding the state of many devices and all their related objects.

MIBs containing trap definitions can be loaded into Foglight. These trap definitions can be configured and grouped in order to produce Foglight alarms. An SNMP trap receiver agent is deployed to a server and listens on a specific port. The agent understands which traps are enabled and how to format them into Foglight alarms based on the mentioned configuration. Managed devices are configured to forward traps to that particular server or port that the agent is deployed on. When a trap is received, the SNMP trap receiver agent, listening on that server or port, forwards enabled traps to the Management Server as alarms. These alarms are fed into Foglight as third-party alarms. Traps that are not enabled can be written to a log or sent in to the alarm browser. This is configured in the Agent Properties. For more information, Configuring Agent Properties for the SNMP Trap Agent on page 32.

Figure 1. Sample alarm



The source for the alarm is the host that initiated the trap. These third-party alarms have additional alarm properties. The key properties are the Technology Monitor which indicates which agent produced an alarm and the trap variables that were available on the trap.

The `sourceId` property contains the trap name if a trap group is not defined. When a trap group is defined, the trap group name is used.

An unformatted trap is a trap that is not defined in Foglight. A trap is defined when the Trap Definition is loaded into Foglight by loading its MIB file. If the trap definition is defined in Foglight, it is no longer considered unformatted. Agent properties display one of the following, when traps are sent in Foglight without a trap definition.

- Undefined_Trap_Creates_Alarm
- Undefined_Trap_Alarm_Duration
- Undefined_Trap_Severity

To format, select the Configuration dashboard from the navigation panel, under Dashboards > Integration > SNMP Trap Administration, and click the **Format** link. For more information about formatting traps, see Configuring Alarms from SNMP Trap Definitions on page 29.

Any variables that the MIB does not recognize is passed in on the alarm with the property name of `Unresolved_Trap_Variable`. Trap variables that the MIB recognizes are preceded with `Variable_` followed by the variable name. For example, `Variable_ifAdminStatus`.

The technology monitor contains the namespace and the agent name used when defining the agent.

For more information, see Understanding the XML Data on page 33 and Loading MIB Definitions on page 28.

External Acknowledgement and Clearing of Alarms Workflow

Foglight for Integrations includes the ability to receive external XML data that can be used to acknowledge or clear alarms existing in Foglight. External systems can format an XML document and feed the ID of an alarm into Foglight to perform this function.

Configuring Foglight for Integrations

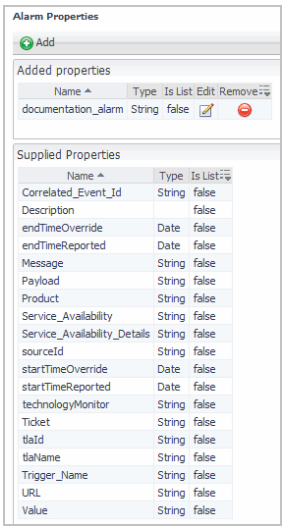
Working in the Alarm Properties Dashboard

To view the alarm properties:

- On the navigation panel under Dashboards, click **Integration** > **Property Administration** > **Alarm Properties**.

  The Alarm Properties view displays the Supplied Properties view and the Added Properties view.

Figure 2. Alarm Properties view



To add an alarm property:

1 On the navigation panel, under Dashboards, click **Integration** > **Property Administration > Alarm Properties**.

2 In the upper left corner of the view, click Add.

The Add Property dialog box appears.

3 In the Name box, type a unique name for the property.

4 In the Type list, click one of the following:

- **String — This** value can include numeric characters, alphanumeric characters, and symbols
- **Boolean — True** or False
- **Integer — A** whole number between the values of -2147483648 and 2147483647
- **Long — A** whole number with a range larger than Integer
- **Float — A** 32-bit floating point number
- **Double — A** 64-bit floating point number

Optional — select the **IS List** check box if you want a property to allow multiple values. Clear the check box if you want to allow only one value.

5 Click Save.

The property is listed in the Added Properties view.

Adding Configuration Item Properties

To add a Configuration Item property:

1 On the navigation panel, under Dashboards, click Integration > Property Administration > CI Properties.

2 In the upper left corner of the view, click Add.

The Add Property dialog box appears.

3 In the Name box, type a unique name for the property.

4 In the Type list, click one of the following:

- **String — This** value can include numeric characters, alphanumeric characters, and symbols
- **Boolean — True** or False
- **Integer — A** whole number between the values of -2147483648 and 2147483647
- **Long — A** whole number with a range larger than Integer
- **Float — A** 32-bit floating point number
- **Double — A** 64-bit floating point number

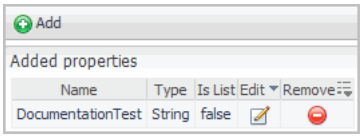
5 Optional — select the **IS List** check box if you want a property to allow multiple values. Clear the check box if you only want to allow one value.

6 Click Save.

Editing Alarm and Configuration Item Properties

To edit a property:

1 In the Added properties view, click the edit icon for a property.



The Edit Property dialog box appears.

2 Edit the properties and click **Save**.

The changes are reflected in the Added Properties view.

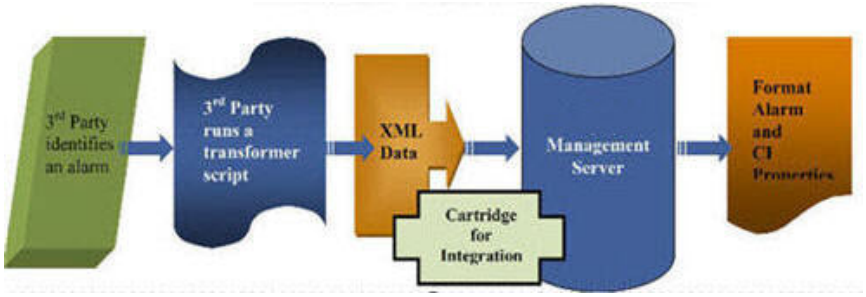
Integration Samples

Integration samples are provided to you as supported samples for integration with Foglight. Other than documented customizations supplied with the samples, Dell does not support customization of the provided samples. Receiving alarms using Foglight for Integrations may have licensing requirements. Consult your Account Representative for details. Also, the source code for the samples is supplied for your convenience.

Receiving Alarms Samples

Quest supports receiving well-formed and valid XML data into Foglight for Integrations (according to the XSD found in the *Integration-Samples.zip* file). Quest does not support derivative works created from the provided samples to generate the XML data.

Figure 3. Alarms from third-parties



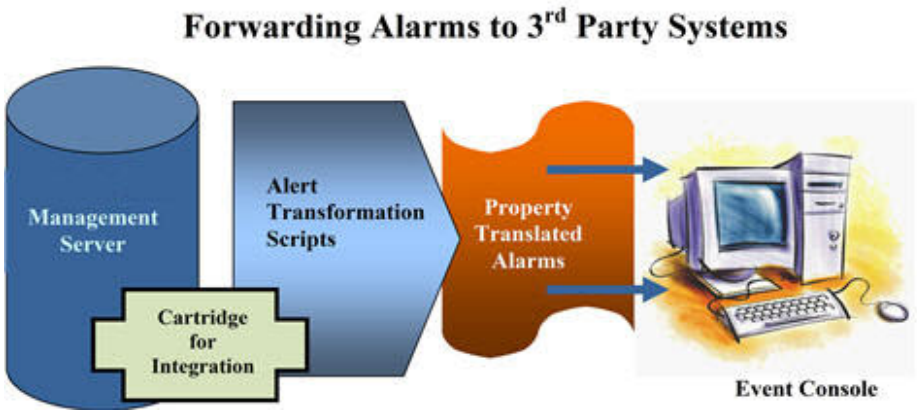
The following receiving samples are available:

- Foglight
- Quest Management Xtensions (QMX)
- Microsoft Operations Manager
- CSVs and Flat Files
- CA Spectrum
- HP OVO
- IBM Tivoli TEC
- Generic Transformer (processes $args)
- Compuware

Forwarding Alarms Samples

Quest supports XML data generated by the Integration action when used in forwarding alarms to other third-party or home grown solutions. Quest does not support derivative works created from the provided samples that use the generated XML data to feed into third-party systems.

Figure 4. Forwarding Alarms to Third-Party Systems



The following forwarding samples are available:

- Quest Management Xtensions (QMX)
- CA Unicenter
- HP OVO
- IBM Tivoli TEC

- SCOM (see the CartridgeForIntegration_SCOMsamplesetup.pdf for setup instructions)

Obtaining Integration Samples

To obtain an integration sample:

1. On the navigation panel, under Dashboards, click **Administration** > **Cartridges** > **Component for Download**.

   The Components for Download page appears.

2. Under the Name column, click **Integration Samples.**

   The File Download dialog box appears.

3. Click **Open** to view the integration samples.

   Optional — Click **Save** to copy the zip file to another location.

4. Unzip the file to view the integration samples and begin working with them.

   Each of the `Integration Samples` sub-directories contains a `Setup.pdf` that contains instructions for establishing that particular feed.

   NOTE: Each of the `Integration Samples` sub-directories contains a `Setup.pdf` that contains instructions for establishing that particular feed.

Receiving Third-Party Alarms in Foglight

Quest supports receiving well-formed and valid XML data into Foglight for Integrations.

Preparing the Data

XML data is used to receive alarms from third-party systems into Foglight. Transformers are provided with the Foglight for Integrations to handle the data appropriately.

XML Elements

Table 1. XML elements and descriptions

XML Element	Description
<configurationItem>	This element identifies that a Configuration Item is being passed in the XML. Attributes can also be attached with the element. They are: **updateMode:** "none/merge/replace" • None — no CI changes • Merge — non-existing properties added, single properties are replaced, and list properties have content merged • Replace — CI contents replaced by data in the XML. This is the default if `updateMode` is not specified. **updateRelationships:** Normally when a Configuration Item is written, it removes all properties and relationships (like the Parent and Child). However, an alarm may come in with Configuration Item updates but without all the relationship information. This setting allows you to update the Configuration Item without affecting any previously defined relationships.
<type>	This *required* element signifies the type of Configuration Item.
<sourceId>	This *required* element should be the unique and persistent ID of the Configuration Item to that technology monitor.
<name>	This required element is the name of the object being passed.
<status>	This required element is the status of the Configuration Item as it relates to the sending Technology Monitor. Valid values are Active and Inactive.
<property>	This optional element is used to pass additional properties about the Configuration Item.
<propertyName>	A valid property as defined in Configuration Item properties administration.
<propertyValue>	A supplied value for the Configuration Item property.
<technicalLevelAgreement>	This optional element identifies that a Technical Level Agreement is being passed in the XML.
<sourceId>	This required element should be the unique ID of the Technical Level Agreement to the technology monitor.
<name>	This required element is the name of the object being passed.
<status>	The required element is the status of the Technical Level Agreement as it relates to the sending Technology Monitor. Valid values are Active and Inactive.
<property>	This optional element is used to pass additional properties about the Technical Level Agreement.
<propertyName>	This element indicates a valid property as defined in Technical Level Agreement properties administration.
<propertyValue>	This element indicates a supplied value for the Technical Level Agreement property.
<alarm>	This element identifies that an alarm is being passed in the XML. Parameters can also be attached. When the severity of an alarm changes, the previous alarm is closed and a new alarm is opened.
<sourceId>	This required element should be the unique id of the alarm to that technology monitor.
<severity>	The *required* element is the severity of the alarm. Valid severity values are: 0-Normal, 2-Warning, 3-Critical, 4-Fatal. Any severities sent in with any other values are mapped to 4 and are displayed in the alarm browser.
<startTime>	The required element is the start time of the alarm.

Table 1. XML elements and descriptions

XML Element	Description
<endTime>	This optional element is the end time of the alarm.
<message>	This *required* element describes the alarm that is occurring.
<property>	This optional element is used to pass additional properties about the object.
<propertyName>	A valid property as defined in Alarm properties administration.

Viewing Third-Party Alarms

Alarms in Foglight indicate significant changes in the state of a monitored resource. An alarm can be generated for a problem, for a resolution of a problem, or the completion of a task. Alarms contain a message that may include details of the data that caused the rule to fire.

To view alarms:

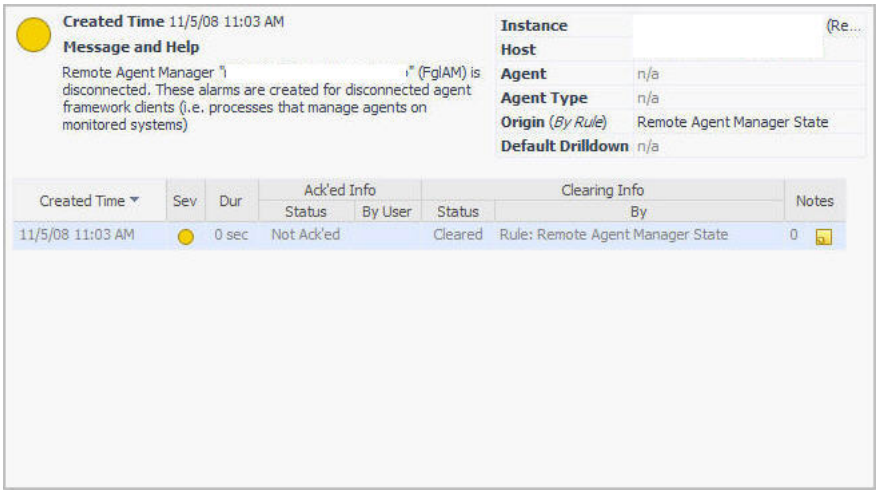
- On the navigation panel under Dashboards, click **Alarms > Alarms**.

To view general alarm details:

- Hover over an alarm in the Alarms view.

  A small dialog box appears.

  Figure 5. Alarm pop-up



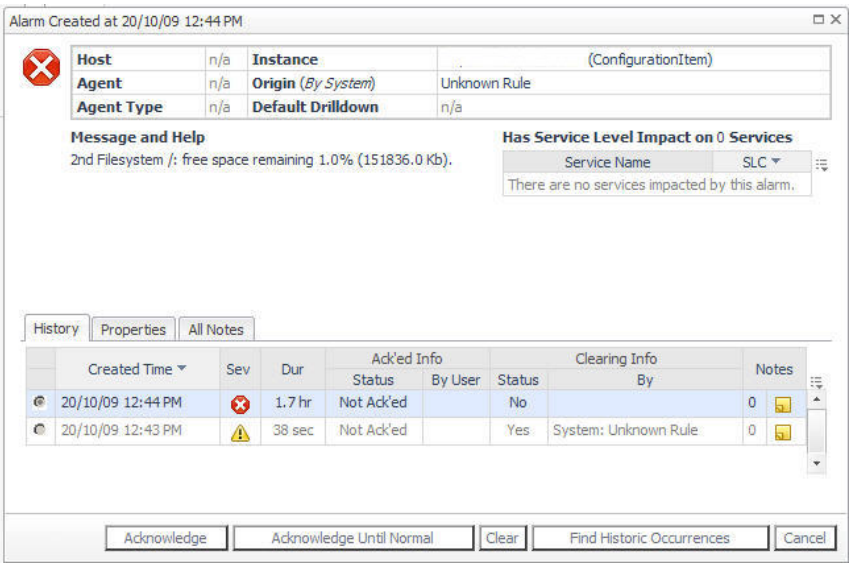
  Third-party alarms contain additional properties, such as start and end times that provide more detailed information about the alarm.

To view more details:

1. Click the alarm listed on the Alarms view.

   The Alarm Detail with Actions information appears.

2. Click the **History** tab.



3 Click one of the following:

- **Acknowledge — if** acknowledged, it indicates that the alarm has been examined. The user name of the person who acknowledged the alarm appears in the Ack'ed By column for that alarm.

  If the operator acknowledges an alarm and then the alarm status changes, the alarm is considered unacknowledged to alert the operator to look a little more closely at the problem.

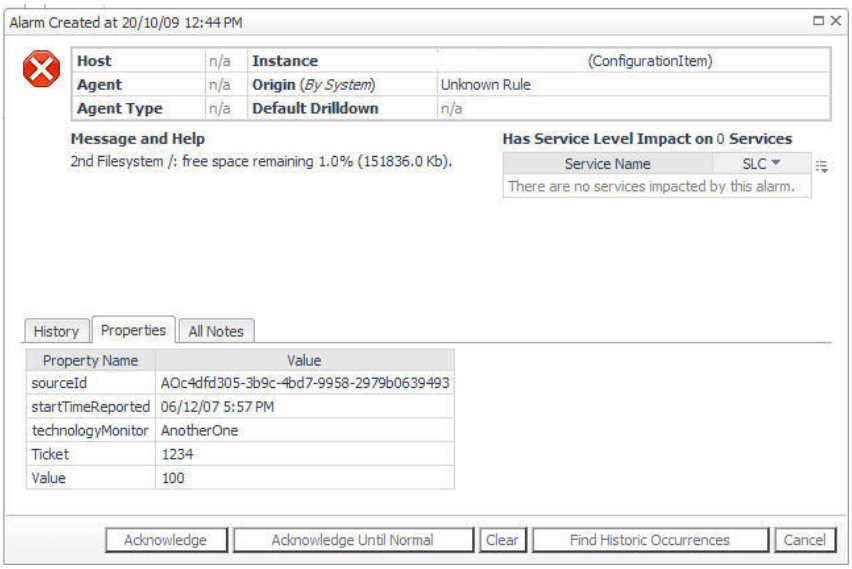
- **Acknowledge Until Normal — this** acknowledges the current alarm and all consecutive alarms that the rule fires on the same instance. This option is available to an outstanding (not yet cleared) alarm only.

  For example, assume that an alarm goes from `Warning` to `Critical` to `Warning` to `Fatal` to `Critical` to `Clear`. The Foglight operator might recognize the alarm at the `Warning` stage as "a problem related to the batch job". If so, the operator might acknowledge the warning, and all subsequent alarms until `Normal`, so anyone else looking at the alarm console knows that they have looked at the problem.

- **Clear — clears** the alarm from the list.

- **Cancel — returns** to the previous view.

4 Click the Properties tab.



The properties tab lists the additional properties that were fed in the alarm. These properties are defined using the **Integration** > **Property Administration** > **Alarm Properties** view.

5 Optional — repeat step 3.

For more information about these actions, see the *Foglight User Guide*.

Filtering Alarms

You can filter the list in the outstanding alarms view using one or more of the following criteria:

- Severity (Undefined, Normal, Fire, Warning, Critical, Fatal)
- Time (range, earliest available, current date)
- Host name
- Source
- Message
- Acknowledged (True or False)
- Agent name

To filter the alarm list:

1 On the title bar of the view, click Alarm Filter Not Set. (If a filter has already been set, the link is titled Alarm Filter Applied.)

The Alarm Filter Not Set/Applied dialog appears.

2 Select or enter your filter criteria.

If you want to use a calendar to choose a date or date range, click the calendar icon to select dates.

The dates that you choose in the calendars display in the From and To fields.

3 Click Apply.

The table refreshes to display the filtered alarm data.

Hiding Columns in the Alarms View

You can hide any of the columns in the Outstanding Alarms view.

To hide columns:

1 Click the edit icon above the table. A pop-up displays a list of the columns in the table.

2 Clear the columns that you want to hide and click Apply. The cleared columns are removed from the table.

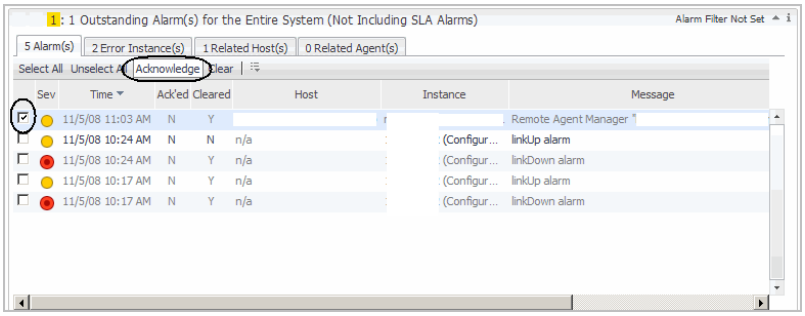
Acknowledging Alarms

The Ack.ed By column in the Outstanding Alarms view indicates whether an alarm has been acknowledged. If an alarm has not been acknowledged, the column is blank. If an alarm has been acknowledged, the column displays the user name of the person who acknowledged it. This information is also stored in an audit report.

To acknowledge an alarm:

1 Click the check box beside the alarm that you want to acknowledge.

2 Click **Acknowledge**.



Clearing Alarms

Alarms from third-party systems may provide start and end times for their alarms. These start and end times fed into Foglight are stored in the user-defined properties of the alarm. By default, any alarm being processed with an end time in the user-defined properties, are not cleared (this occurs when the `Auto_Clear_Integration_Alarms_with_EndTime` registry variable is set to false).

In certain instances, a third party may send an alarm with an end time in the future. This occurs when the technology monitor is unable to provide an indication that an alarm condition is no longer occurring. However, the third-party systems check again and send another alarm if the condition is still occurring. For example, a condition is checked every 15 minutes. If at 12:00 A.M. the condition is true, an alarm is sent with a start time of 12:00 A.M. and an end time of 12:15 A.M.. The end time is 12:15 A.M. because the condition is checked again at that time and if still true, another alarm is sent.

In Foglight, we may not want to automatically clear the alarm because the user would never see the alarm in the browser and know that it is occurring. We also do not want the user to have to manually clear all these alarms.

To automate the clearing of future dated alarms, the registry variable, `Auto_Clear_Integration_Alarms_with_EndTime`, can be changed. The default value is false. When true, the alarm with a user-defined end time is cleared automatically when the alarm is first received. When set to false, any alarm with a user-defined end time is not cleared in that manner. Otherwise, these must cleared using the user interface or by using the new Clear Integration Alarm Rule. This new rule is disabled by default. When enabled, this rule runs every five minutes. This rule looks for third-party alarms with an end time in the past that have not been cleared, and then clears the alarm. This keeps the alarm in the browser until the end time is reached.

NOTE: If the registry variable is set to true, the Clear Integration Alarm Rule does not not close any alarms because the alarm processing would have already closed any alarm with an end time.

Changing the Registry Variable

1. On the navigation panel, under **Dashboards**, click **Administration** > **Rules and Notifications** > **Manage Registry Variables**.
2. Click `Auto_Clear_Integration_Alarms_with_EndTime`.

   The Manage Registry Variables view appears.
3. Edit the registry variable.

   For more information on how to edit registry variables, see the *Foglight Administration and Configuration Guide*.

Processing Alarms

The processing of third-party alarms manages the updates made to the user-defined properties.

Table 2. Alarms

Type & Severity	Severity Change	Alarm Scenario	Alarms Behavior	Actions Fired
New Warning, Critical or Fatal (2 - 4)	No	Single alarm received with no endtime. Severity is anything from 2 to 4	1 alarm created.	1 action fired. The creation of a new alarm.
New Warning, Critical or Fatal (2 - 4)	No	Single alarm with a start and endtime.	1 alarm created. Alarm is cleared if registry variable specifies to do so.	1 or 2 actions fired. The first action is for the alarm creation. If the registry variable specifies to clear alarms with end times, then a second action is fired.
New Normal (0)	No	Single alarm received with no endtime. Severity is 0, which both map to Normal.	Ignored	None
Update Warning, Critical or Fatal (2 - 4)	No	Update alarm comes in to update user-defined data on existing non-cleared alarm. Severity is the same as previous alarm.	Updates to user-defined data on original alarm.	1 action fired. Update user-defined data.
Update Warning, Critical or Fatal (2 - 4)	Yes	Update alarm comes in to update user-defined data on existing non-cleared alarm. Severity is different from previous alarm.	Closes out previous alarm. Creates alarms. Old alarm user-defined data remains unchanged.	2 actions fired. Closing old. Alarm creation of new alarm.
Update Warning, Critical or Fatal (2 - 4)	No	Update alarm comes in that closes existing non-cleared alarm. New alarm has the same severity as original alarm.	User-defined data on original alarm updated. Alarm cleared if registry variable set.	1 or 2 actions fired. One for the update of user-defined data. One for the clear if the registry variable is set.
Update Warning, Critical or Fatal (2 - 4)	Yes	Update alarm comes in that closes existing non-cleared alarm. New alarm has a different severity than original alarm.	Clears previous alarm, creates new non-cleared alarm. Alarm properties stored on new alarm only. New alarm is cleared if registry variable is set. Previous alarm user-defined data untouched.	2 or 3 actions fired. The first two actions clear the old alarm, and creates an alarm. The optional third is if the new alarm is automatically cleared per the registry variable setting.

Table 2. Alarms

Type & Severity	Severity Change	Alarm Scenario	Alarms Behavior	Actions Fired
Update Normal (0)	Any	Update alarm that sets severity to 0, which both map to Normal.	Clears previous alarm. New incoming alarm data is discarded.	1 action fired. Old alarm clear.
UI — Acknowledge	No	User acknowledges alarm in the browser.	Alarm is marked as acknowledged.	1 action fired. Acknowledgment of the alarm.
UI — Clear	No	User clears alarm in the browser.	Alarm is marked as cleared and no longer shows up in the browser.	1 action fired. Clearing of the alarm.

Working with Configuration Items

Any monitored object coming from an external source is considered a configuration item. A Configuration Item is anything that can be monitored. For example, a host, a database, a memory component, or a CPU. The hierarchy is not pre-defined. If an event consolidator sends an event tracking an interface on a switch, a Configuration Item is created for that interface. If the Event Consolidator sends information about the switch and the relationship between the switch and the interface, then two Configuration Items are created: one for the switch and one for the interface. In addition, the “detail” property (which is inherited in the model) of the switch is set to the interface, allowing Foglight to understand that some Configuration Items are “parents” to other Configuration Items.

Each Configuration Item is unique. It has a combination of technology monitor and sourceId. The number of Configuration Items that are created depends on the types of systems sending alarms into Foglight. If two different technology monitors manage the same Configuration Item (a server for example), two different Configuration Items are created. If a technology monitor manages a Configuration Item which is the same as a device Foglight manages (and appears in the Host model), it creates a unique Configuration Item.

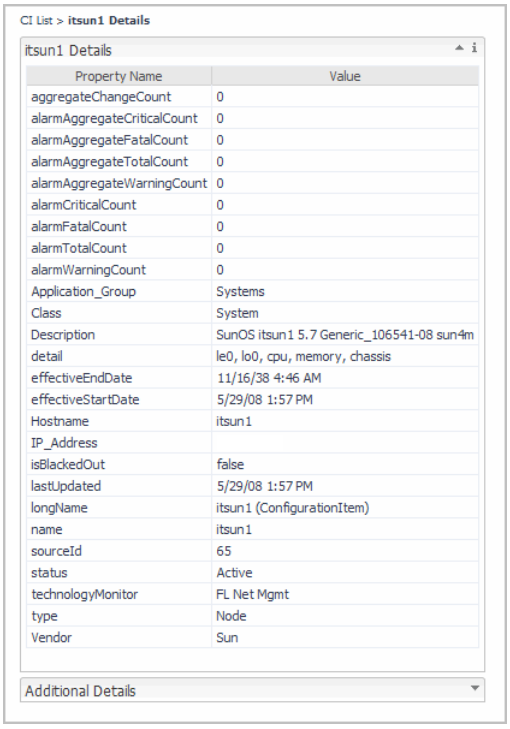
Foglight for Integrations provides the Foglight Administrator the capability of defining CI Properties using the CI Property view. In this view, the Administrator can define a new property along with Type and Is List options.

NOTE: In a federated environment, it is important to keep CI Properties consistent between federated Management Servers. For example, if one federated Management Server has a property called External asset id that is a type of `Integer` and `Is List`, other federated servers should have a property called External asset id that is a type of `Integer` and `Is List`. CI Properties must be consistent across federated servers.

Reviewing the Property Details of a Configuration Item

To review general details of a Configuration Item:

1. On the navigation panel, under Dashboards, click **Integration** > **CI List**.
2. Click a Configuration Item listed in the CI List view.



For details on the state of the Configuration Item, click the down arrow on the **Additional Details** view.

Filtering Configuration Items

To filter Configuration Items:

1 On the top right corner of the Configuration Items view, click CI Filter.

The CI Filter dialog box appears.

2 Set the filtering criteria.

The CI List is filtered by Node, by default.

Table 3. CI List properties and descriptions

Property	Description of required properties
Name	A unique name of a Configuration Item. Use Regex — use regular expressions.
Technology Monitor	The third-party system that sent the object. Use Regex — use regular expressions.
Type	The type of Configuration Item. For example, Node, Interface, Database, and so on. The filter is defaulted to Type=Node. Use Regex — use regular expressions.
Status	The status of the Configuration Item in the source system. The status can be either Active or Inactive.

Table 3. CI List properties and descriptions

Property	Description of required properties
Match effectiveStart Date	Date the Configuration Item was created. **From**: date and time **To**: date and time For example, 5/30/08 and time 4:28:34. Click the calendar icon to select dates.
Earliest	If the entered Created date is not correct, you can select this check box to locate the earliest available date.
Current Date	Filter using the current date.

3 Click Find.

A list of Configuration Items using that criteria appears in the view.

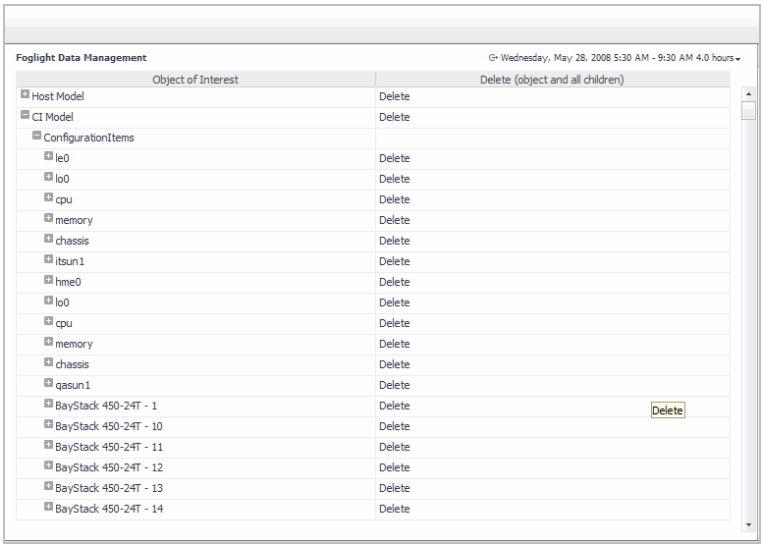
- Click Reset to remove the filtering criteria and enter new information.
- Click Clear to remove the filtering criteria. The complete list of Configuration Items is listed in the view.

Deleting Configuration Items

To delete a Configuration Item from the CI List:

1 On the navigation panel, under Dashboards, click **Foglight** > **Servers** > **Foglight Data Management**.

2 Under the Object of Interest column, click **CI Model** > **ConfigurationItems** to expand the list.



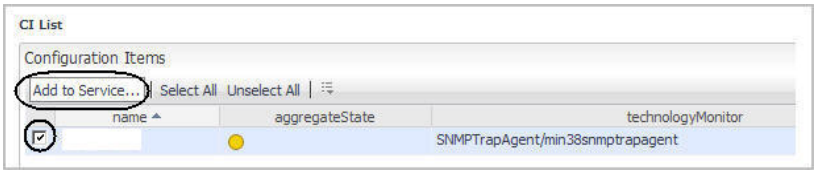
3 Click **Delete** on the corresponding row of a Configuration Item to delete the object and all related children.

Adding a Configuration Item to a Service

1 On the navigation panel, under Dashboards, click **Integration** > **CI List**.

2 Select a Configuration Item check box located in the left column of the table.

3 Click **Add to Service**.



4 Select one or more services for the Configuration Item.

5 Click **Add**.

Using the CI Model in Service Builder

The Service Builder dashboard provides the functions needed to create a service and edit existing services. When you create a service, a corresponding service level is automatically created. The Service Builder dashboard allows the user to group of one or more components.

Services are used as inputs on many other dashboards (Hosts Table, Agents) besides the Services dashboards, as well as in reports. Defining a good set of services can make other dashboards more useful and easier to understand.

For more information about how to create, edit, and remove services, see the *Foglight User Guide*.

Selecting a Configuration Item in a Service Builder

The third-party Configuration Items can be used in building services. This can be done by selecting All Models and then opening the CI Model. This shows all Configuration Items fed from third-parties. These can be selected and included in the service. For example, accounts payable may want to include Configuration Items in a service because it is possible Foglight is not monitoring them, but rather by a third-party system.

To select all Configuration Items fed from third-party systems:

1 In the Service Builder, create a service.

For more information about how to create a service, see the Foglight User Guide.

2 On the Service Builder view, click All Models.

3 Select the Configuration Items you want to add to the service.

4 Click Add.

All Configuration Items fed from third-party systems appear in the view. These can be selected and included in the service.

Properties of CI Model:

Value — The value of the property.

Data Type — The data type is a data-object template, and it determines the structure of a data object. Examples of data types are Host, AppServer, WebLogic, WebSphere, Agent, and Event.

Forwarding Alarms to Third-Party Systems

You can forward any alarm into another third-party system. A new action, called *IntegrationAction*, has been introduced to allow the user to forward these alarms.

NOTE: For Microsoft System Center Operations Manager (SCOM) integration rule setup, see the CartridgeForIntegration_SCOMsamplesetup.pdf. For all other integration samples, see Forwarding Alarms on page 23.

NOTE: A sample Alarm Integration rule is provided when Foglight for Integrations is installed. It is disabled by default. Once enabled, it can be modified to the specific requirements needed to forward alarms to a third-party system.

Forwarding Alarms

NOTE: This section pertains to all integration samples except the Microsoft System Center Operations Manager (SCOM) integration sample. For the SCOM sample, see the CartridgeForIntegration_SCOMsamplesetup.pdf.

To forward an alarm to a third-party system:

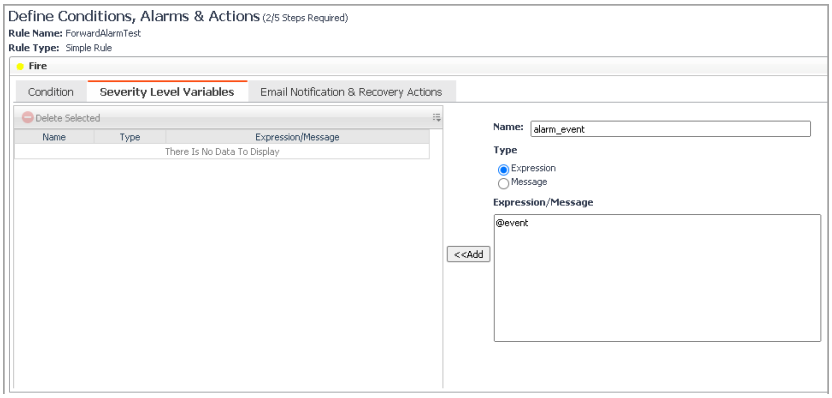
1. On the navigation panel, under Dashboards, click **Administration** > **Rules & Notifications** > **Create Rule**.

   The Create Rule view appears.

2. Click the **Rule Definition** tab, and in the **Rule Name** dashboard, type a name.
3. Click to select the **Simple Rule** option button and the **Event Driven** option button, then select **AlarmSystemEvent** from the Event Name drop down.

4. Set the **Rule Scope** to **No Scoping Query**.
5. Click the **Severity Level Variables** tab to add the rule variables *alarm_event* and *script* and set the following:
   a. In the **Type** field, click to select the **Expression** radio box to add the `alarm_event` variable.
   b. In the **Name** box, type `alarm_event` and in the **Expression/Message** box type `@event`.

      The event object is passed to the output script.



c Click <<**Add**.

d In the **Type** field, click to select the **Message** radio box to add the `script` variable.

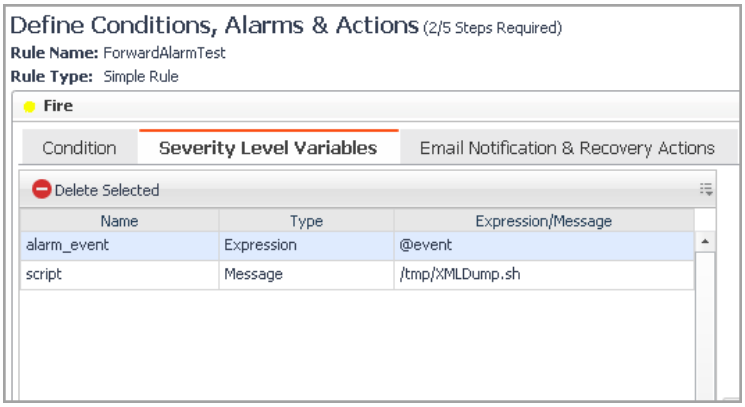
e In the **Name** box, type `script` and in the **Expression/Message** box type the path of the script you want to use. For example, `/tmp/XMLDump.sh`.

This script is replaced with the script being used to feed the third-party system.

> **NOTE:** On the Windows platform, if running a Perl script, the Expression/Message must begin with “perl” followed by the full path of the script being run. For example: perl c:\dell...\script_name.pl

> **TIP:** On the navigation panel, under Dashboards, click Administration > Cartridges > Components for Download, and then click Integration Samples. Save or Open the file and locate the script called `sample_alarm_parser.pl`. This script is an XML parser that parses the outgoing XML data. It can be used as a template to start building your own script.

f Click <<**Add**.



6 Click the **Condition** tab and click the **Fire** heading.

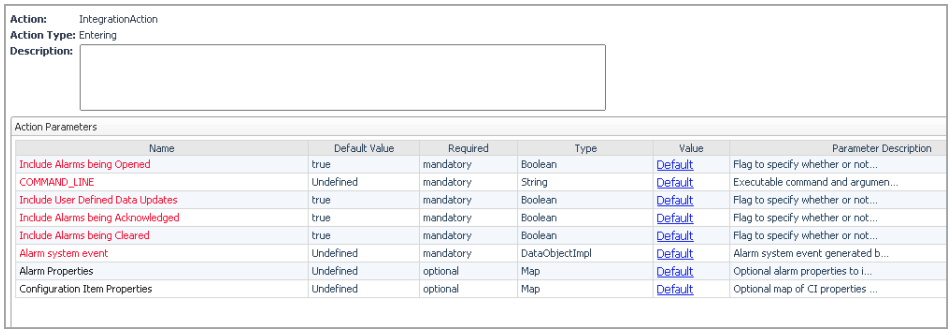
a In the **Condition** box, type `true` and then click the **Test Rule Logic** button.

b Click the **Email Notification & Recovery Actions** tab, and set **Action Type** to **Entering.**

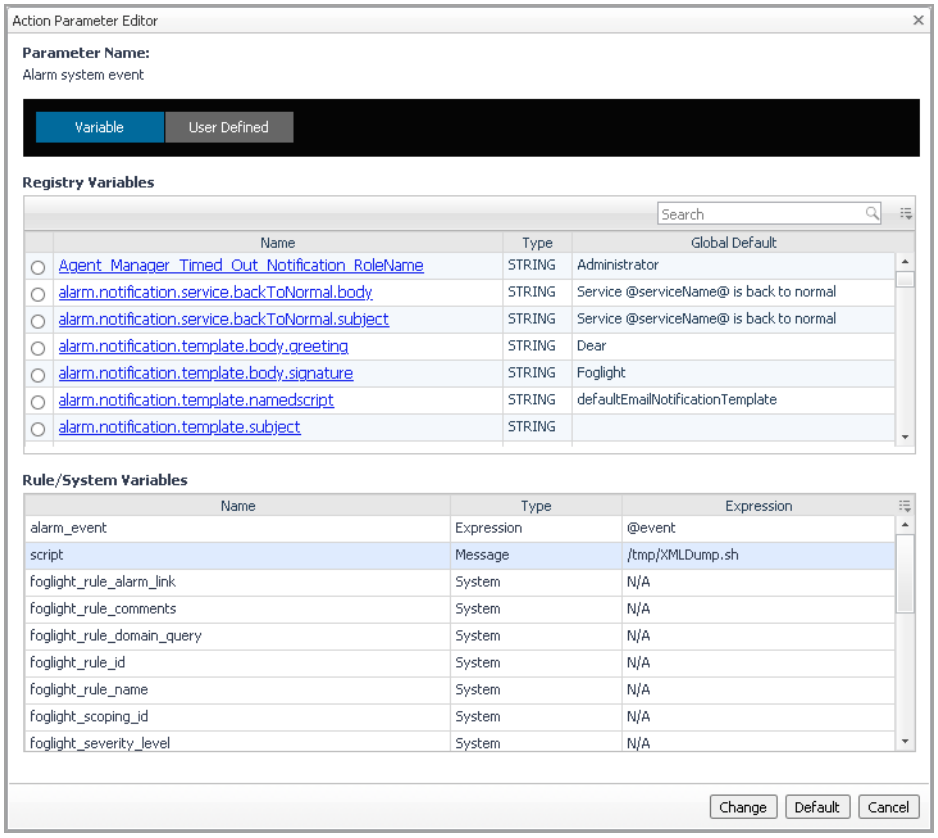
Entering is the default value.

c In the **Action** list, click **IntegrationAction**.

d Click <<**Add**.

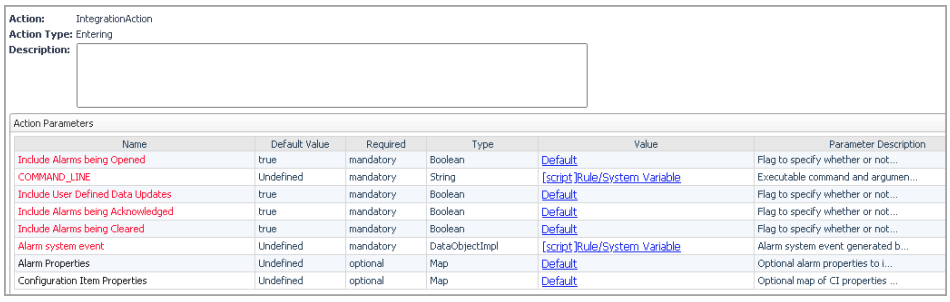


e For `alarm system event`, click **Default** and select **alarm_event** from the **Rule/System Variables**.



f Click **Change**.

g For `COMMAND_LINE`, click the **Default** link, select **script** from the **Rule/System Variables** and click **Change**.



Four action parameters listed in the preceding image.

- **Include Alarms being Acknowledged**
- **Include Alarms being Cleared**
- **Include Alarms being Opened**
- **Include User Defined Data Updates**

These are used to control what alarm updates trigger this action. The value can be either true or false.

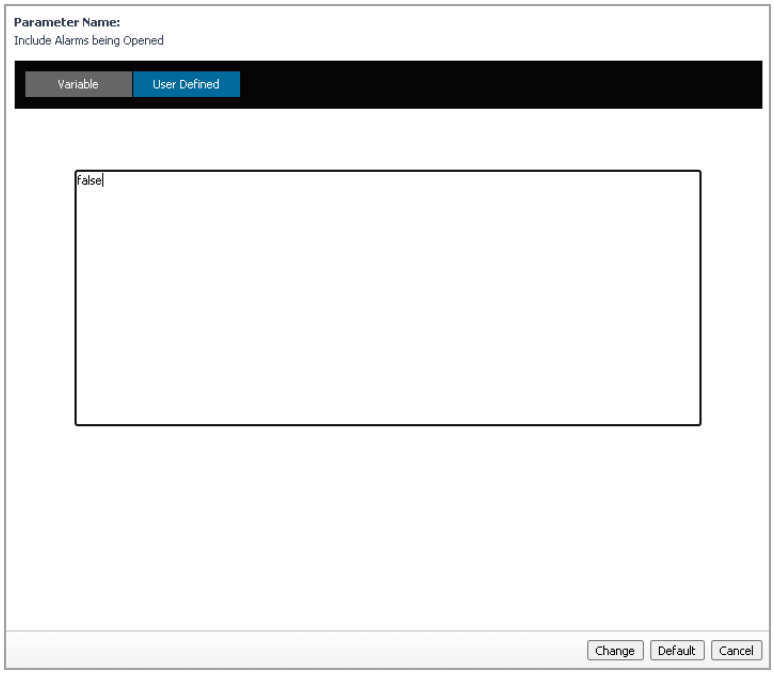
When these are set to true, and when they occur, they trigger the integration action.

7 Click **Finish** to save the rule.

To change the value to false:

1 Click on the default link next to each to change.

2 In the Action Parameter Editor, click the **User-Defined** tab and type `false` to no longer trigger the action when that update occurs.

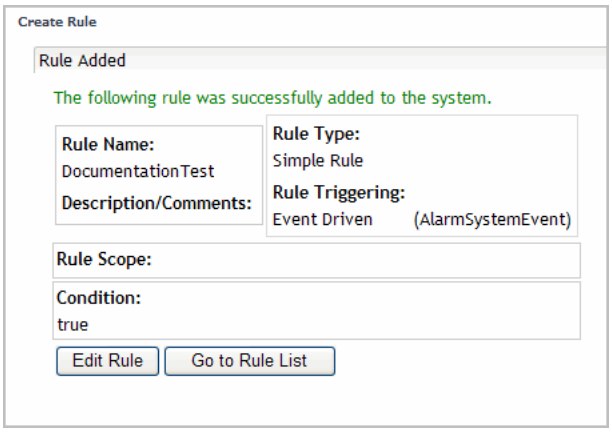


3 Click the **Change** button to save the changes.

4 To set to **true,** click **Default**.

5 Click **Go To Action List** go back to **Rule Editor**, click **Next** and then click **Save** to save the rule.

For more information about rules, see the *Foglight Administration and Configuration Guide*.



For more information about rules, see the *Foglight Administration and Configuration Guide*.

Creating an XMLDump Script

This script takes Foglight Alarm XML data and dumps it into the `/tmp` directory. This would be replaced with a script used to transform the XML data into the format that the receiving third-party system requires.

To create an XML Dump script:

1 Create a script and type the following:

```
#!/bin/sh
cat > /tmp/dumpit.out.$$
echo "made it out!"
```

2 Save this script to the /`temp` directory.

XML Information

The XML data generated contains information about the alarms as they exist in Foglight.

- For Foglight generated alarms, the `ruleId`, `ruleName`, and `sourceName` are added to the XML data.
- For all alarms, the `uniqueId`, `createdTime`, `clearedTime`, and `acknowledgedTime` element data are provided in the XML data output.
- The `type` element data on outgoing Configuration Items is set to the Foglight topology `type` for Foglight generated alarms.
- For Third-party alarms, the `type` is from the type property that is passed in on the Configuration Item.
- All properties of the alarm or topology objects are passed on the outgoing feed. If the property is another object, only the name of the object (for example, severity) is used.

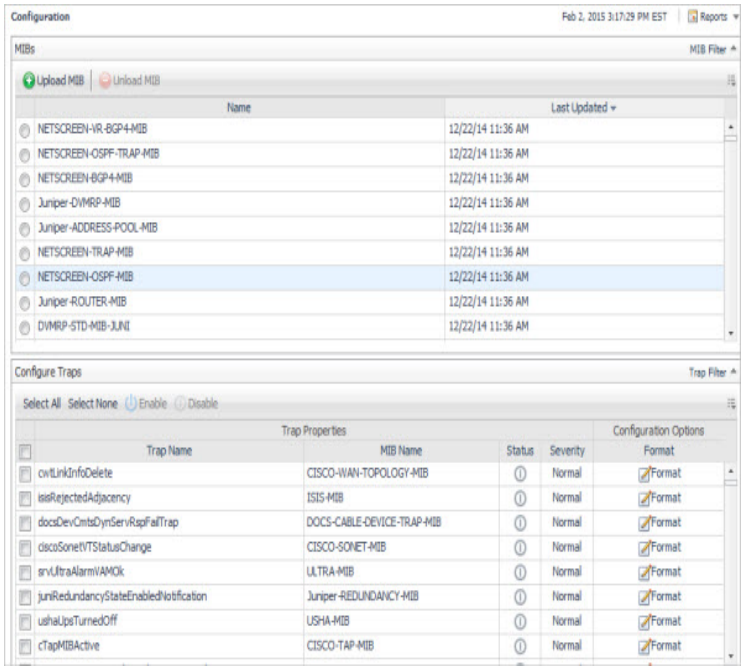
Receiving SNMP Traps

Foglight for Integrations provides the capability to load Management Information Base (MIB) files. These MIB files contain trap definitions. The trap definitions define the traps that can be received and their variables. In Foglight, the trap definitions are used to configure the subsequent alarms that are sent to Foglight.

Loading MIB Definitions

To upload a MIB, navigate to the MIB Trap Configuration dashboard found on the navigation panel under **Dashboards**, and click **Integration** > **SNMP Trap Administration** > **Configuration**.

Figure 6. Configuration view



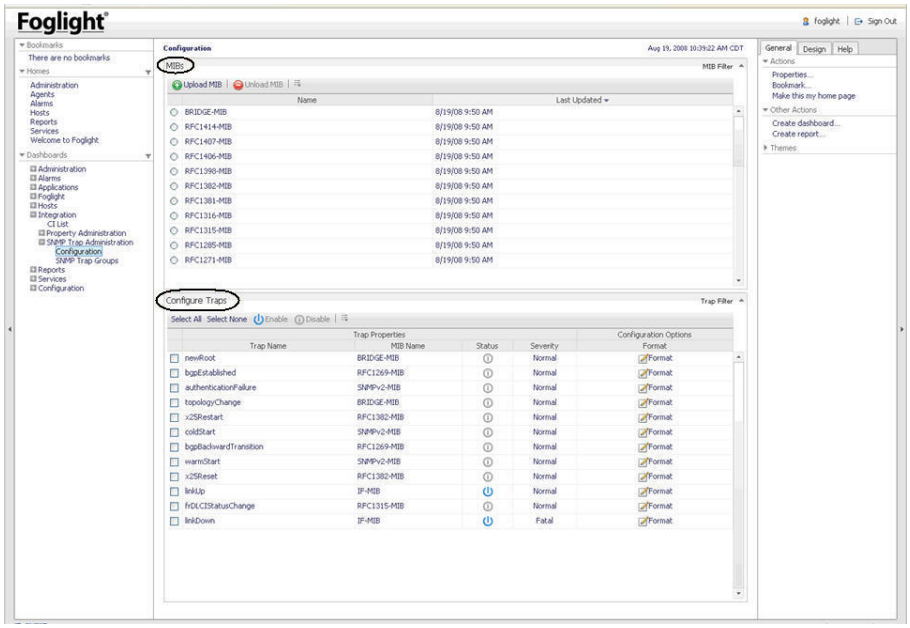
This page lists all loaded MIBs in the upper view. It contains all traps in the lower view. The MIB view identifies the MIB name and the timestamp when the MIB was loaded.

Click the Upload MIB button to begin the upload process. You are prompted for the location of the MIB file. Once supplied, click the Apply button to load the MIB. The results of the load are displayed in the upper view of the Configuration dashboard.

NOTE: Certain MIBs rely on other MIBs for additional definitions. Prerequisite MIBs must be loaded first, then subsequent MIBs, otherwise an error occurs. When removing MIBs, Foglight for Integrations detects if a MIB is a prerequisite to another MIB, and disables the Unload MIB button.

MIBs for many vendors can be found on the navigation panel, under Dashboards, and by choosing Administration > Cartridges > Components for Download. Click the Other MIBs link to download a compressed file. Once uncompressed, notice under each vendor directory the corresponding MIB files and an `order.txt` file that indicates the order in which the MIBs need to be loaded.

When a MIB is loaded that contains a trap definition, the trap shows up in the lower view and is disabled by default.



To enable trap definitions:

1 On the navigation panel, under Dashboards, click **Integration** > **SNMP Trap Administration** > **Configuration.**

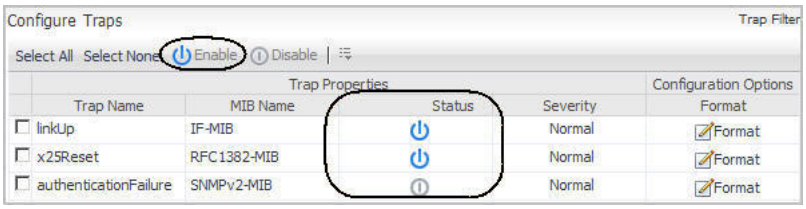
A list of trap definitions associated with the loaded MIBs appears in the lower view.

2 On the Configure Traps view, select the check boxes for each trap you want to load.

Optional — to quickly select all configuration traps, click **Select All**.

3 Click **Enable**.

The Status icon updates.



NOTE: Once the trap is enabled, configure it for creating an alarm in Foglight. By default, the severity is set to Normal. Alarms with a normal severity do not show up in the event browser.

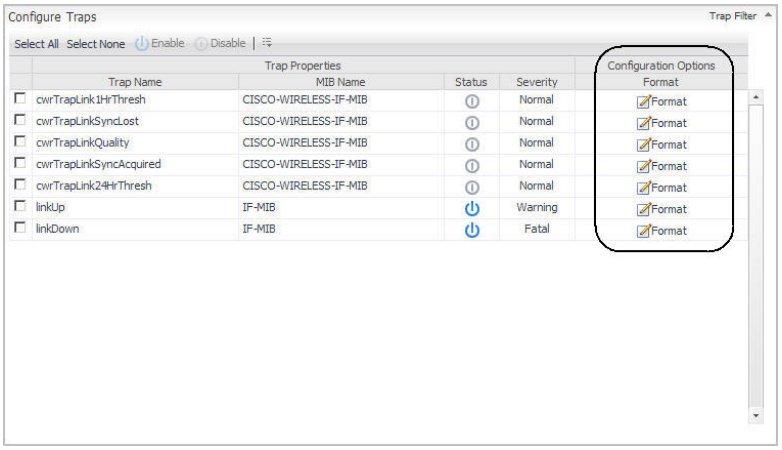
Configuring Alarms from SNMP Trap Definitions

Trap Definition configuration is used to format the alarm that is sent into Foglight.

To format traps:

1 On the navigation panel, under Dashboards, click **Integration** > **SNMP Trap Administration** > **Configuration**.

2 On the Configure Traps view, select **Format** for a Trap you want to configure.



The Edit Trap view appears.

3 Set any of the following options:

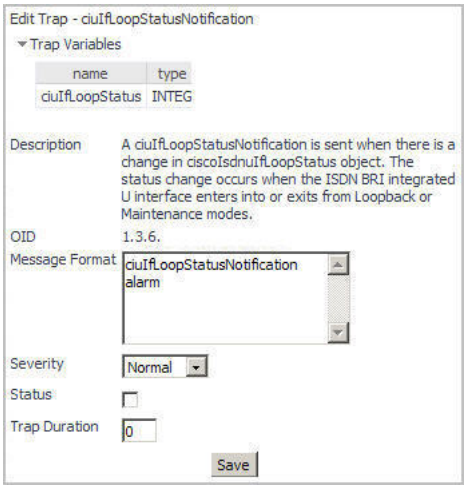


Table 4. Format options and descriptions

Format Option	Description
Trap Variable	Trap variables are defined in the trap definition and passed in when a trap is received. These are sent as properties on the alarm. They can also be included in the message for the alarm.
Description	The Description is defined in the trap definition and passed in when a trap is received. This is sent as a property on the alarm. It can also be included in the message for the alarm.
OID	The Object Identifier (OID) of the trap as defined in the MIB file.
Message Format	The message format that is provided on the alarm sent into Foglight based on the trap information. Trap variables can be specified in the message. For example, if the trap variable `ifIndex` is included in the message, it would be written `${ifIndex}`. Likewise, the description from the trap definition can be included in the message by using `${description}`. You can use the variables `${host}` `${trapname}` and `${technologyMonitor}` in the message format. The `${host}` is where the trap came from.
Status	The status of a trap. The status can be either Enabled or Disabled.
Trap Duration	Forces an end time on the event sent into Foglight. The end time would be the start time of the trap with the duration added to it. Setting the Trap Duration to zero would keep the trap alarm open until it clears.

Grouping SNMP Traps

You can group SNMP traps together when they are monitoring the state on the same object. In this case, one trap may indicate an impact on an object, and another trap may indicate the correction of an issue on that object. These traps need to work together to portray the correct state of an object.

To create an SNMP Group:

1 On the navigation panel, under Dashboards, click **Integration** > **SNMP Trap Administration** > **SNMP Trap Groups**.

All trap groups are listed.

2 Click **Create.**

3 Type an SNMP Trap Group Name and then click **Save**.

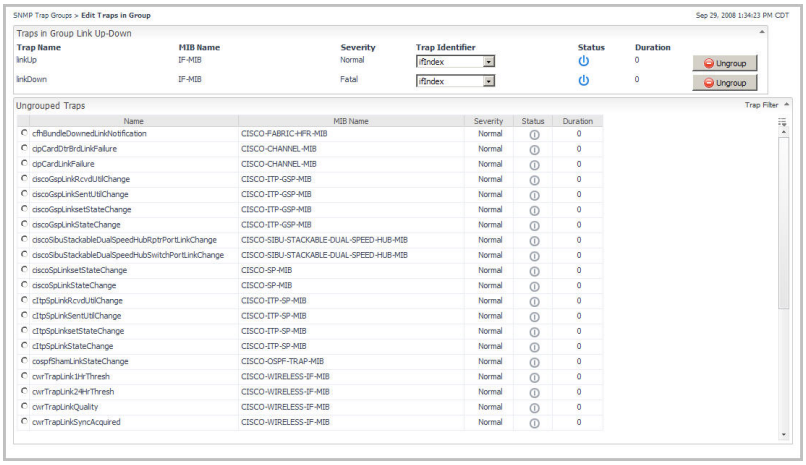
To edit an SNMP Trap Group:

1 On the navigation panel, under Dashboards, click **Integration** > **SNMP Trap Administration** > **SNMP Trap Group**.

This presents two views. The first view lists all the traps that have currently been assigned to the group. The lower view contains a list of all the traps that are available.

The upper view lists all traps in the trap group and the variables associated with each trap. Variables can be selected to make the trap apply to a unique component on the device. For example, for Link Up and Link Down traps, if the `ifIndex` variable is not selected, a Link Up alarm on interface may close a Link Down alarm on interface. By selecting the `ifIndex`, the index would have to be the same for the alarms to clear one another.

In the lower view, checking the option button next to a trap assigns the trap to the group. When checked, the trap moves to the upper view.



2 In the Ungrouped Trap view, click the option button for each trap you want to add to the Trap Group.

To remove a trap from an SNMP Trap group:

1 On the navigation panel, under Dashboards, click **Integration > SNMP Trap Administration > SNMP Trap Groups**.

2 Click **Edit** for the appropriate SNMP Trap Group.

3 On the *Traps In Groups* view, in the upper right portion of the display area, click **Ungroup**.

After removing SNMP Traps, the metadata and variable information are removed from their associated Alarms and the SNMP Configuration listing.

The *Ungrouped Traps* view updates to include the removed trap groups.

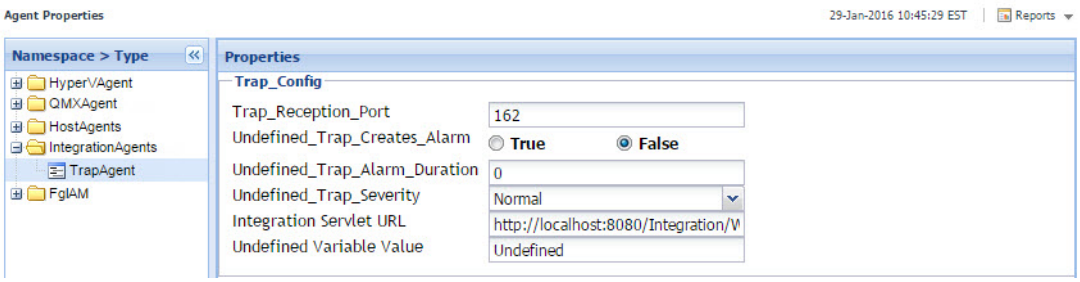
Configuring Agent Properties for the SNMP Trap Agent

The SNMP Trap Agents can be configured based on the parameters listed below.

To update agent properties:

1 On the navigation panel, under **Dashboards**, click **Administration** > **Agents** > **Agent Properties**.

2 On the **Namespace** > **Type column**, click **IntegrationAgents** > **TrapAgent**.



3 Set any of the following properties:

Trap_Reception_Port — Traps are configured to send to this specific port on the server the agent is deployed.

Undefined_Trap_Creates_Alarm — This is a Boolean value that indicates how to handle unformatted traps. If false, the unformatted trap is logged and discarded. If true, an alarm is sent to the alarm browser. Traps are logged in the agent log. The agent log can be retrieved through the UI using the Agent Status dashboard. The agent can be selected and the Get Log button can be clicked.

Undefined_Trap_Alarm_Duration — When an unformatted trap is sent to the alarm browser, the duration of the trap is noted. When combined with the Clear Integration Alarm Rule, the trap could stay open in the event browser.

Undefined_Trap_Severity — The severity of the alarm for the unformatted trap.

Integration Servlet URL — The URL of the Management Server (FMS) that receives the trap from the agent. Change this from *localhost* to the name or IP address and port number of the Management Server. For example: `http://123.456.78.9:8080/Integration/Write.`

Undefined Variable Value — The value for this property is used in the message format when a trap variable that was configured in the message format does not come in on the trap. It is defaulted to `Undefined`.

For more information about Agent Properties, see the *Foglight Administration and Configuration Guide*.

Troubleshooting

What happens to traps that I do not have a MIB loaded for?

They are recognized as unformatted traps. There is a configuration option in the Agent Properties dashboard for the SNMPTrapAgentCartridge UI to report these as alarms. They can also be logged.

Where do I configure the duration for unformatted traps?

There is a configuration option in the Agent Properties for the SNMPTrapAgentCartridge UI.

What happens if I load a MIB more than once?

A MIB cannot have items removed from it. Any MIB loaded a second time only adds new symbols or changes some existing symbols.

Can I load a file that contains more than one MIB?

Yes, but only the first MIB definition is recognized. Split the MIIBs into separate files and reload them individually.

How do I configure an SNMP port?

Set the parameters in the Administration > Agents > Agent Properties dashboard.

What port should the Foglight Agent Manager run on?

In order to start a trap receiver agent, the Foglight Agent Manager must be installed. It should be installed as ROOT if you want to run it on the default port of 162. If run as NON-ROOT, you must configure a port higher than 1024 in the agent properties (Administration > Agent > Agent Properties).

Receiving Acknowledged and Cleared Commands

XML data is used to receive Acknowledge and Clear commands from third-party systems.

Understanding the XML Data

The following XML elements are utilized.

Table 5. XML elements

XML Element	Description
clearAlarms	This is the element that lists the alarms to be cleared (one or more).
id	When beneath the `clearAlarms` element or the `acknowledgeAlarms` element, this is the id of the alarm as assigned in Foglight. This id would originally be passed to the external system from Foglight in the outgoing XML. The field in that XML is called `<uniqueId>`.
acknowledgeAlarms	This is similar to the `clearAlarms` element but is used to acknowledge alarms.
clearAlarm	This element is used to clear a single alarm. The alarmId is passed using an attribute in the format `id="xxx"`.
acknowledgeAlarm	This element is similar to the `clearAlarm` element, but is used to acknowledge a single alarm.

Sample of XML

<?xml version="1.0" encoding="UTF-8"?>

<alarmActions xmlns="http://www.quest.com/xml/ns/foglight/ integration/alarmActions"

```
xmlns:xsi="http://www.w3.org/2001/ XMLSchema-instance"
xsi:schemaLocation="http://www.quest.com/xml/ns/foglight/ integration/alarmActions
../src/META-INF/alarm_actions.xsd">
<clearAlarms>
<id>e696b857-11f0-4eb6-b3db-8542dc34f38a</id>
</clearAlarms>

<acknowledgeAlarms>
<id>f554t857-11f0-4eb6-b3db-8542dc34f38a</id>
</acknowledgeAlarms>

<clearAlarm id="o987y651-11f0-4eb6-b3db-8542dc34f38a"/>
<acknowledgeAlarm id="f987a651-11f0-4eb6-b3db-8542dc34f38a"/>
</alarmActions>
```

Viewing Alarms

To view alarms, access the Alarms dashboard. For more information about viewing alarms, see the *Foglight User Guide.*

Deploying and Configuring QMX Agents

The QMX Agent can be deployed and used to receive data from Quest Management Extensions (QMX). QMX instrumentation packs can be configured to send data to Foglight. When configured, the packs send data to the `FL_Performance_Event` MOF table. The QMX Agent can be deployed to a server and then subscribe to that table to get the performance data.

Deploying QMX Agents

The QMX agent can be located either directly on the same server as QMX or on a remote server that has RPC and WMI access to the QMX server. Both installations are supported, but the collocated agent is easier to diagnose since the only network traffic that takes place is between two machines instead of three. A remotely located agent needs to communicate remotely with Management Server and remotely with the QMX server.

Configuring QMX Agent Properties

The agent properties for the QMX agent contain only a single secondary ASP where each of its rows defines another QMX host to poll data from. The definition for each column is as follows:

Table 6. QMX agent properties

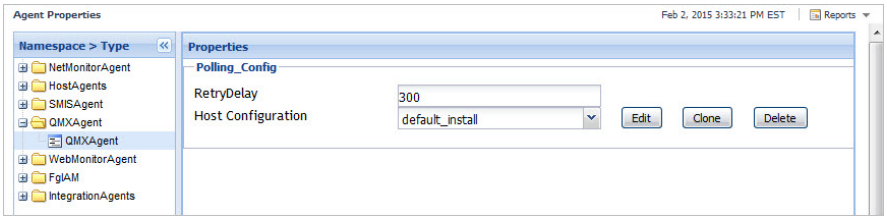
Agent Property	Description
Host	The host name or IP address of the QMX host. This should be set to localhost if the agent and QMX are collocated, otherwise set to the host name of the QMX machine.
Username	The Windows user name that has access to poll WMI information from the specified host. **NOTE:** Administrative access is required by default for polling this information. Setting up other users is outside the scope of this document.
Password	The password associated with the specified user name.
Domain	If the username is part of a domain, it can be specified here. This field can usually be left blank.

To update agent properties:

1 On the navigation panel, under Dashboards, click **Administration > Agents > Agent Properties.**

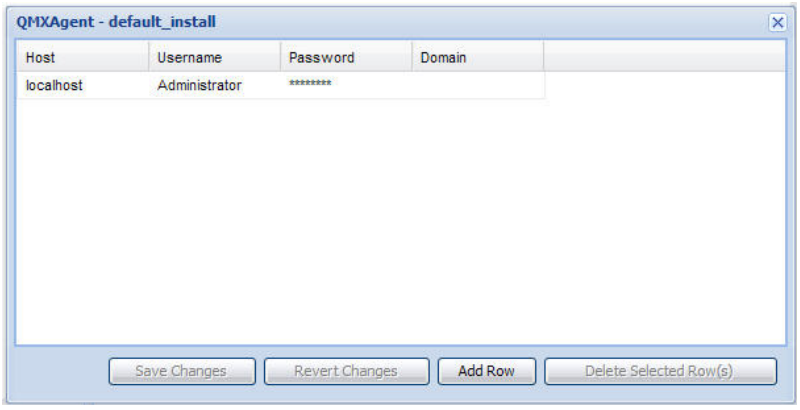
2 On the Namespace > Type column, click **QMXAgents** and then click **QMXAgent** type.

3 Click **Edit**.



The running QMX host is listed in the table.

4 Edit the **Username** and **Password**.



For more information about Agent Properties, see the *Foglight Administration and Configuration Guide*.

Starting the Agent

Once the agent is configured, it can then be activated using the standard Foglight Agent activation process.

NOTE: QMX must be installed before activating the agent. If QMX is not installed, the agent fails to activate correctly.

You do not need to run QMX to activate the agent because the agent does not interface directly with QMX.

Configuring QMX

Requirements

In order for QMX to publish Foglight data to the QMX agents, the following requirements must be met:

- A Foglight capable instrumentation pack must be installed in QMX.
    a Confirm a successful installation of this instrumentation pack by looking for a Foglight tab within that instrumentation pack's configuration menu. If present, the installation was successful.
    b For Instrumentation Packs that are Foglight enabled, start any of the QMX Virtual Agents for that Instrumentation Pack to send data to the QMX agents.

Configuration

To configure QMX:

For QMX Versions earlier than 4,0,0,499:

1 To configure QMX to send data to Foglight, the file `FoglightTranslationTable.xml` file must be copied to the `\eXc Software\WMI Providers\nonWindows\Virtual Agent Library\MOM` directory on the QMX server. The file is located in the Administration > Cartridges > Components for Download dashboard.

2 Set the following values in the `commonroutinesInsertPerformanceRecord.js` script:
    a Open `\eXc Software\WMI Providers\nonWindows\Virtual AgentLibrary\MOM\CommonRoutinesInsertPerformanceRecord.js;`
    b Set `l_booluseSDK = false and l_booluseFL = true`.

3 Click the MOM GlobalVariables tab, and uncheck the **g_boolUseOpsMgrSingleton** variable.

For QMX Versions 4,0,0,499 and later:

1 Update `\WMI Providers\nonWindows\Virtual Agent Library\MOM\GlobalVariables.js.`

2 Set `variable g_boolIsFogLight` to `true` on line 51.

Verification

Once data is being published using the QMX agent into Foglight, new hosts for each Virtual Agent appears within the Foglight Hosts view.

Troubleshooting

To troubleshoot any problems with an agent, create both a Foglight and Foglight Agent Manager (FglAM) support bundle. The QMX virtual agent log is also typically required to ensure that the QMX data is being published.

Configuring ServiceNow Integration

Foglight ServiceNow Integration allows using ServiceNow API to create ServiceNow incidents for Foglight alarms.

Requirements

Using Foglight ServiceNow Integration requires a ServiceNow account with ServiceNow roles:

- incident_manager
- personalize_choices

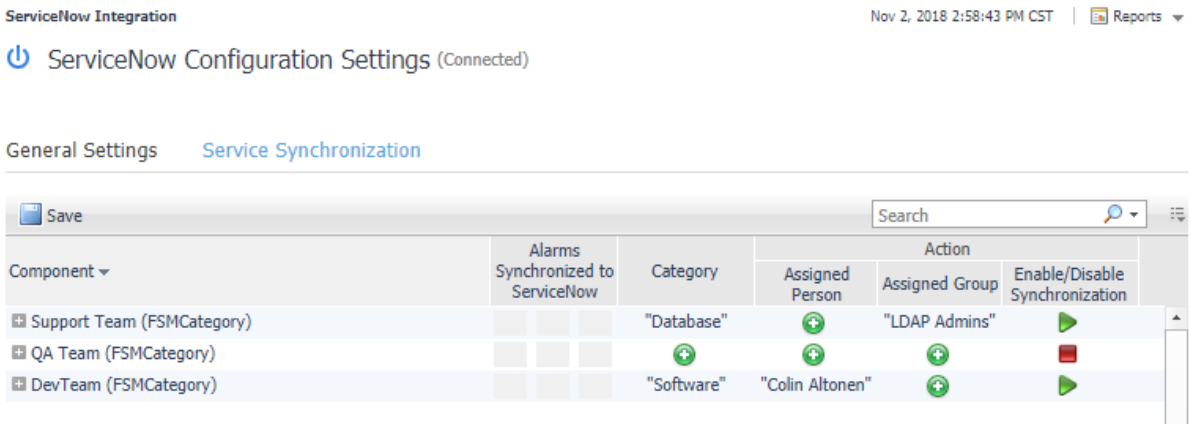
Configuring ServiceNow integration

To connect to ServiceNow instance:

1. Select Dashboard **Administration** > **Integration** > **ServiceNow Integration** to open ServiceNow configuration dashboard.
2. Click **Get Started** to start the configuration wizard.
3. In the first wizard screen, **ServiceNow Instance Account**, connect to the ServiceNow instance by providing the ServiceNow URL and credential.
4. Click **Next** and Foglight will verify the provided ServiceNow instance URL and credential. If the verification passes, the Summary page will be displayed.
5. Then click **Finish** and the ServiceNow Integration main Dashboard will be displayed, which includes **General Setting** tab and **Service Synchronization** tab.
    a. **General Setting** tab displays current ServiceNow Integration settings and status.
    b. **Service Synchronization** tab lists all Foglight services and allows user to configure service based alarm to incident synchronization.

Configuring service based synchronization

Foglight ServiceNow supports service based alarm synchronization, users can configure service based alarm synchronization in **Service Synchronization** dashboard:



To configure service synchronization:

1 Find the Foglight service or child service you want to operate in service table.

2 Click the **Category** column for the selected service, and select the ServiceNow **Category** of the alarms you want to synchronize to.

3 Click **Assigned Person** OR **Assigned Group** to select the default assignee for all synchronized ServiceNow incidents.

4 Click **Enable/Disable Synchronization** to enable synchronization for the selected service.

5 Click the **Save** button to save all changes, this step is required because the configuration only works after it is saved.

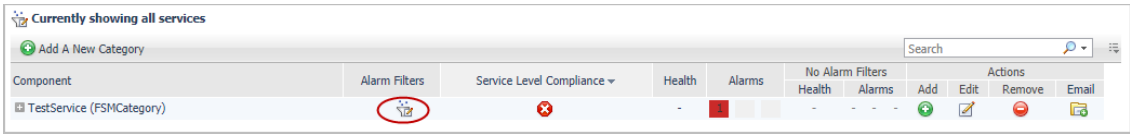
If you want to keep some irrelevant alarms from falsely causing a service outage, define an alarm filter in the Service Builder.

To define an alarm filter:

1 Select **Dashboard Services** > **Service Builder**.

2 Click the **Alarm Filters** to open the Alarm Filters dialog box.

Figure 7. Open Alarm Filters dialog box

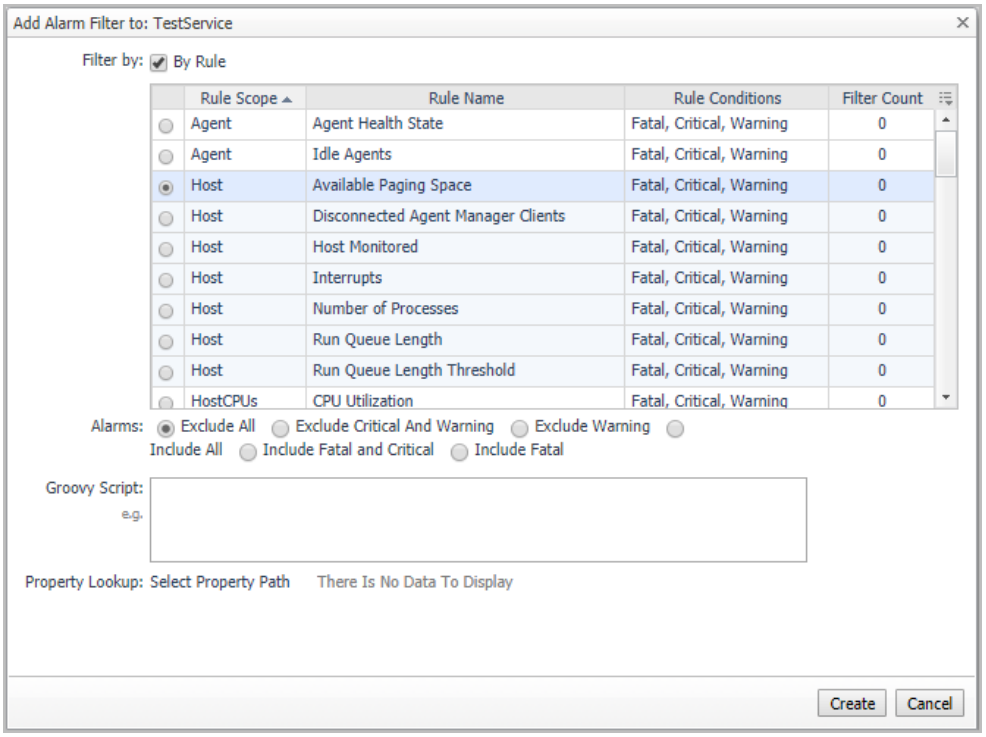


3 Click **Add** in the Alarm Filters dialog box. The Add Alarm Filter dialog box appears.

Figure 8. Add Alarm Filter dialog box

4 Then, select one or more filter options:

Figure 9. Select filter options



- **By Rule**—select the By Rule check box and choose a rule for which its alarms are to be included or excluded. You can define more than one alarm filter for the same rule.
- **Alarms**—select the types of alarms to include or exclude.

  NOTE: If both Include and Exclude filters are defined, Foglight includes alarms that are specified in the Include filters as long as they are not excluded by the Exclude filters. If only Include filters are defined, all alarms are excluded except those specified in the Include filters.If only Exclude filters are defined, all alarms are included except those specified in the Exclude filters.

- **Groovy Script**—refine the filtering by running an optional groovy script. The current alarm is the only parameter passed to this script, meaning you can filter on anything that is referenced by the alarm. An example of a groovy script is:

  ```
  @alarm.get('topologyObject').getType().getName() ==
  'Windows_System_System_Table'
  ```

5 Click **Create**.

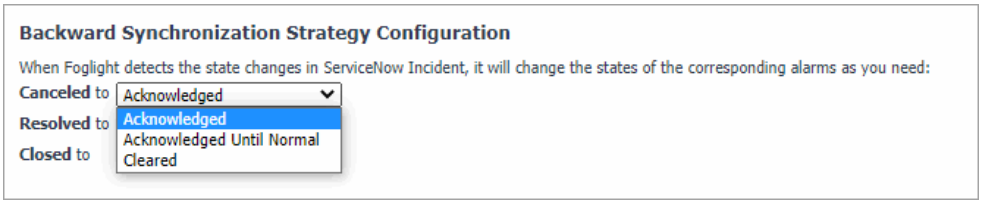
The filter is saved on the Alarm Filters list and the number is displayed to represent the number of filters created for the selected component.

After applying the Alarm Filters, irrelevant alarms will not be synchronized to ServiceNow.

Configuring backward synchronization strategy

Foglight ServiceNow Integration helps Foglight user to synchronize FMS alarms to ServiceNow incidents.

Figure 10. Backward Synchronization Strategy Configuration



The Backward Synchronization Strategy Configuration allows users to synchronize ServiceNow incidents status changes to Foglight. After applying the configuration, when Foglight detects the state changes in ServiceNow Incident, it will change the states of the corresponding alarms basing on the configuration.

2

Reference

Review these topics if you are unfamiliar with views and rules in Foglight for Integrations.

Views

Foglight displays monitoring data in views that group, format, and display data. The following topics describe the main types of views.

Dashboards are top-level views that do not receive data from other views. Dashboards usually contain several lower-level views. The dashboards supplied with Foglight, and dashboards that users create, are available in the navigation panel.

Lower-level views in Foglight can be added to dashboards or can be accessed by drilling down from a dashboard. They receive and display data directly from the Management Server or from other views. Some views filter or select data that appears in other views in the same dashboard. Some are tree views with expandable nodes for selecting servers, applications, or data.

CI List View

Purpose

The CI List view allows you to quickly access detailed information on the Configuration Items that have been fed into Foglight from third-party systems. In addition, from this view you can add a configuration item to a service.

This view is located in the navigation panel of the Foglight user interface, under the Integration dashboard.

For more information about working with configuration items in this view, see Reviewing the Property Details of a Configuration Item on page 19.

Figure 11. CI List view displaying all configuration items

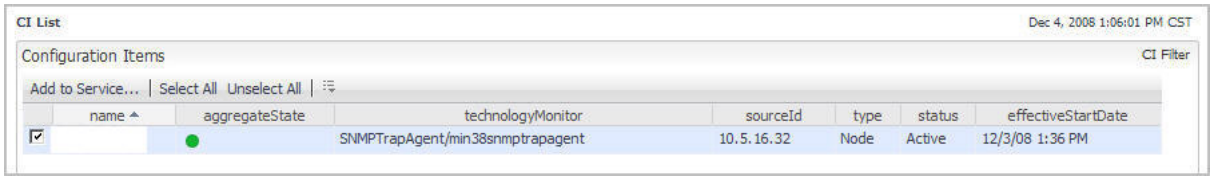


Figure 12. CI List node details view

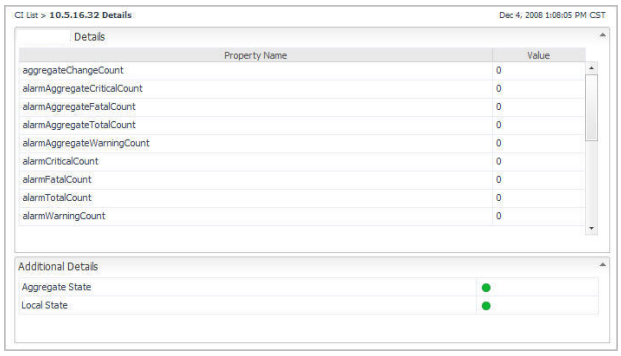


Table 7. Configuration Items fields and descriptions

Data Type	Description
name	A unique name of a Configuration Item.
aggregateState	The current state of the Configuration Item. For more information, see Alarm Properties View on page 42.
technologyMonitor	The technology monitor that sends Configuration Items.
source Id	A value that uniquely identifies the Configuration Item in the technology monitor.
type	A simple description of the type of object. For example, Node, Interface, Database, Business Service, and so on.
status	The status of the Configuration Item in the source system. The status can be either Active or Inactive. State icons indicate the status of a domain, server, application, or process.
effectiveStartDate	The date and time the Configuration Item was created.

Table 8. State icons

Icons	Description
	Normal
	Warning
	Critical
	Fatal

Alarm Properties View

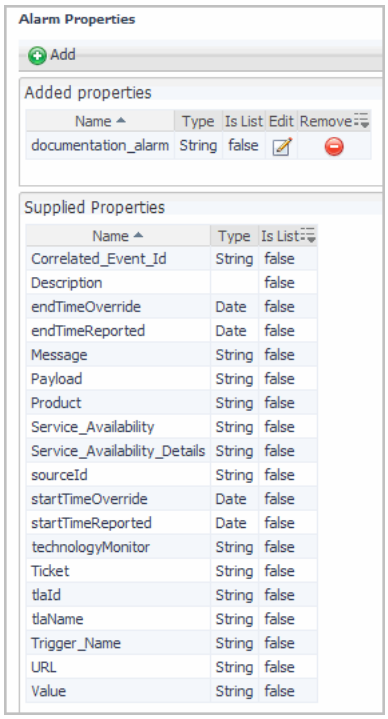
Purpose

The Alarm Properties view lists added and supplied properties for alarms. Added Properties lists unique properties that the user or operator has added for an alarm. Supplied Properties is a list of the properties that are supplied by default.

Use this viewto add properties for an alarm, as a result of that customizing an installation to meet the needs of the customer. Users can define a (limited) data type for the property along with a flag indicating whether many values are supported.

This view is located in the navigation panel of the Foglight user interface, under the Integration > Property Administration dashboard.

Figure 13. Alarm Properties view



For more information about how to add alarm properties, see Working in the Alarm Properties Dashboard on page 8.

Table 9. Added properties and Supplied Properties

Data Type	Description
Name	A required field that assigns a unique alarm property name.
Type	**String:** This value can include numeric characters, alphanumeric characters, and symbols **Boolean**: True or False **Integer:** A whole number between the values of -2147483648 and 2147483647 **Long:** A whole number with a range larger than Integer **Float**: A 32-bit floating point number **Double:** A 64-bit floating point number
Is List	True — Set the property to allow multiple values False — Set the property to allow only one value

CI Properties View

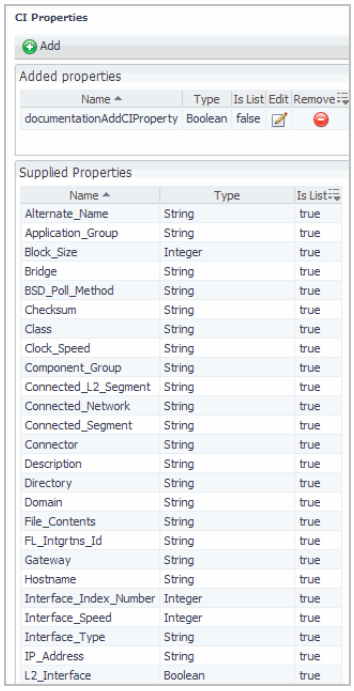
Purpose

The CI Properties view lists all the properties that can be included on a Configuration Item. Supplied Properties are provided by default when the cartridge is installed. The properties that users add are Added Properties. Once defined, properties can be included in the XML for a Configuration Item.

This view allows you to add properties for a Configuration Item, so you can customize an installation to meet the needs of the customer. Users can define a (limited) data type for the property, along with a flag indicating whether many values are supported.

This view is located in the navigation panel of the Foglight user interface, under the Integration > Property Administration dashboard.

Figure 14. CI Properties view



For more information about how to add configuration item properties, see Adding Configuration Item Properties on page 9.

Table 10. Added properties and Supplied Properties

Data Type	Description
Name	A required field that assigns a unique Configuration Item property name.
Type	**String: C**an include numeric characters, alphanumeric characters, and symbols **Boolean:** True or False **Integer:** A whole number between the values of -2147483648 and 2147483647 **Long**: A whole number with a range larger than Integer **Float**: A 32-bit floating point number **Double:** A 64-bit floating point number
Is List	True — Set the property to allow multiple values False — Set the property to allow only one value

Configuration View

Purpose

Use the Configuration view to quickly review the MIBs loaded in Foglight. The MIBs table and the Configure Traps table are displayed in this view.

The MIBs table lists the name of the MIB and the date the MIB was last updated. MIBs can be uploaded using the Upload MIB button. When updating, the user is prompted to locate the MIB file. Any errors found while loading MIBs is presented in the user interface.

In the Configure Trap table, a list of subsequent trap definitions associated with the loaded MIBs appears. This table provides a list of traps and their associated MIB name. It also includes the status indicating if the trap is enabled or disabled and the severity of the traps. The Format link allows you to format the alarm information being fed into Foglight.

This view is located in the navigation panel of the Foglight user interface, under Integration > SNMP Trap Administration > Configuration dashboard.

Figure 15. Configuration view

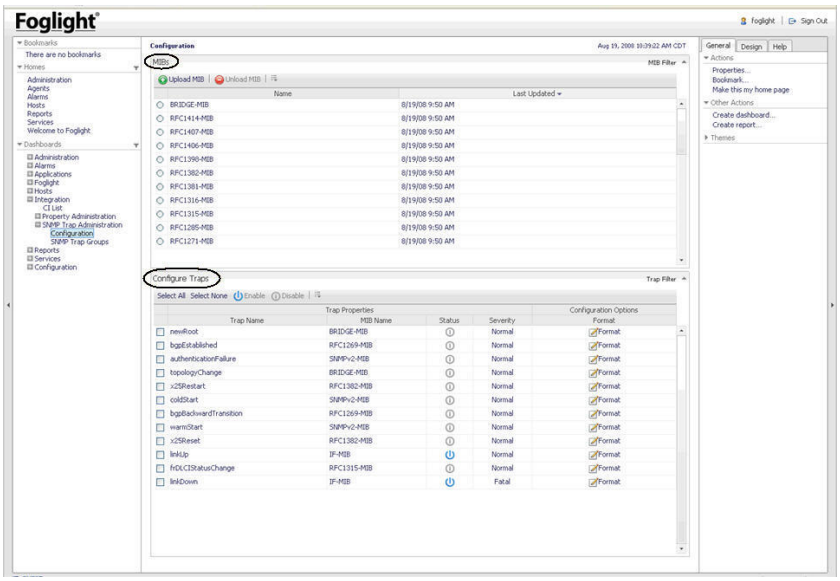


Table 11. Configure Trap Properties

Property	Description
Trap Name	The name of the trap that is provided in the MIB file used to load the trap.
MIB Name	The name of the MIB file that contained the trap.
Status	The status of a trap. The status can be either Enabled or Disabled.
Severity	The severity that is given to an alarm that is created when this trap is received.
Format	The configuration options for a Trap. Trap Variables: Trap variables are defined in the trap definition and passed in when a trap is received. They are sent as properties on the alarm. They can also be included in the message for the alarm. Description: The Description is defined in the trap definition and passed in when a trap is received. It is sent as a property on the alarm. It can also be included in the message for the alarm. OID: The OID of the trap as defined in the MIB file. Message Format: The message format that is provided on the alarm sent into Foglight based on the trap information. Trap variables can be specified in the message. For example, if the trap variable `ifIndex` is included in the message, it would be written `${ifIndex}`. You can use the variables `${host}` `${trapname}` and `${technologyMonitor}` in the message format. The `${host}` is where the trap came from. Trap Duration: Forcing an end time on the event sent into Foglight. The end time would be the start time of the trap with the duration added to it. Setting the Trap Duration to zero keeps the trap alarm open until it clears.

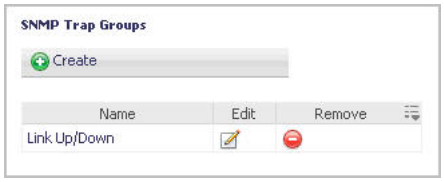
For more information about how to work with SNMP traps, see Receiving SNMP Traps on page 27.

SNMP Trap Groups View

Purpose

The SNMP Trap Group view allows you to group related traps together in order to manipulate alarms associated with the state of an object. Once a trap group is added, you can then add traps to the trap group.

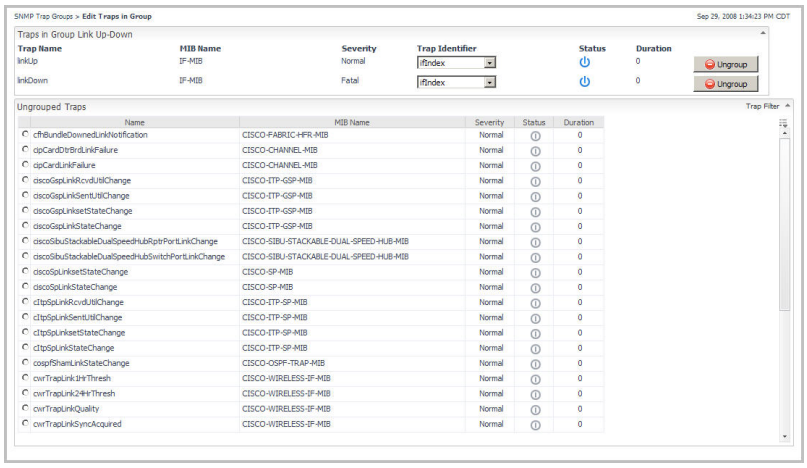
Figure 16. SNMP Trap Groups view



If you choose to edit an SNMP Trap Group, a view of the Traps In Group table and the Ungrouped Traps table is displayed. The Traps In Group table is used to assign variables to make the traps specific to an object. The Ungrouped Traps table lists all traps that were loaded in Foglight. The trap can be included in the trap group by selecting the box next to the trap. When selected, the trap is removed from the Ungrouped Trap list and placed in the Traps in Group table.

This view is located in the navigation panel of the Foglight user interface, under Integration > SNMP Trap Administration > SNMP Trap Groups.

Figure 17. Edit Traps in Group view



For more information about how to work with SNMP traps, see Grouping SNMP Traps on page 31.

Rules

Foglight allows you to create flexible rules that can be applied to complex, interrelated data from multiple sources within your distributed system. You can associate several different actions with a rule, configure a rule so that it does not fire repeatedly, and associate a rule with schedules to define when it should and should not be evaluated.

Different types of data can be used in rules, including registry variables, raw metrics, derived metrics, and topology object properties.

There are two types of rules in Foglight: simple rules and multiple-severity rules. A simple rule has a single condition, and can be in one of three states: *Fire*, *Undefined*, or *Normal*. A multiple-severity rule can have up to five severity levels: *Undefined*, *Fatal*, *Critical*, *Warning*, and *Normal*.

Rule conditions are regularly evaluated against monitoring data (metrics and topology object properties collected from your monitored environment and transformed into a standard format). Therefore, the state of the rule can change if the data changes. For example, if a set of monitoring data matches a simple rule's condition, the rule enters the *Fire* state. If the next set does not match the condition, the rule exits the *Fire* state and enters the *Normal* state.

A rule condition is a type of expression that can be true or false. When it evaluates to true, the rule is said to fire, causing any actions that are associated with the rule or severity level to be performed. You can configure a rule to perform one or more actions upon entering or exiting each state. When a multiple-severity rule fires, an alarm also appears in Foglight.

See "Introduction to Rules" and "Creating and Editing Rules" in the Foglight Administration and Configuration Guide for more information.

Alarm Integration Rule

Purpose

The Alarm Integration rule is used to forward alarms out of Foglight into a third-party system.

Although disabled by default, when enabled, it will fire for all alarms out of the box and will generate an XML document that can be fed to a script. The script defined in the action can be used to send to the third-party system.

The rule triggering is event-driven.

This rule can be viewed by opening the navigation panel, and under Dashboards, click **Administration** > **Rules & Notifications** > **Manage Rules** and then click on a **Alarm Integration** rule.

For more information about how to customize and edit the Alarm Integration rule, see Forwarding Alarms to Third-Party Systems on page 23.

Clear Integration Alarm Rule

Purpose

The Clear Integration Alarm rule is used to clear alarms from third-party systems that have been closed by the third-party system. This rule should be used to clear alarms when the *Auto_Clear_Integration_Alarms_with_EndTime* parameter is set to false and the users do not want to manually clear alarms closed by the third-party systems.

This rule can be viewed by opening the navigation panel, and under Dashboards, click **Administration** > **Rules & Notifications** > **Manage Rules** and then click on a **Clear Integration Alarm Rule**.

For more information about how to edit and customize the Clear Integration Alarm rule, see Clearing Alarms on page 17, and Changing the Registry Variable on page 17.

Incident Integration Rule

Purpose

The Incident Integration rule forwards incidents from Foglight to a third-party system. This rule is disabled by default. When this rule is enabled, all incidents are generated into XML. This XML can be fed into the a script which defines the actions used to send incidents to a third-party system. This rule is incident driven.

This rule can be viewed by opening the navigation panel, and under Dashboards, click **Administration** > **Rules & Notifications** > **Manage Rules** and then click on a **Incident Integration Rule**.

For more information about how to edit and customize the Incident Integration rule, see Forwarding Incidents in the Foglight Cartridge for Integration User Guide.

About Us

Quest creates software solutions that make the benefits of new technology real in an increasingly complex IT landscape. From database and systems management, to Active Directory and Office 365 management, and cyber security resilience, Quest helps customers solve their next IT challenge now. Around the globe, more than 130,000 companies and 95% of the Fortune 500 count on Quest to deliver proactive management and monitoring for the next enterprise initiative, find the next solution for complex Microsoft challenges and stay ahead of the next threat. Quest Software. Where next meets now. For more information, visit https://www.quest.com/.

Technical support resources

Technical support is available to Quest customers with a valid maintenance contract and customers who have trial versions. You can access the Quest Support Portal at https://support.quest.com.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request.
- View Knowledge Base articles.
- Sign up for product notifications.
- Download software and technical documentation.
- View how-to-videos.
- Engage in community discussions.
- Chat with support engineers online.
- View services to assist you with your product.