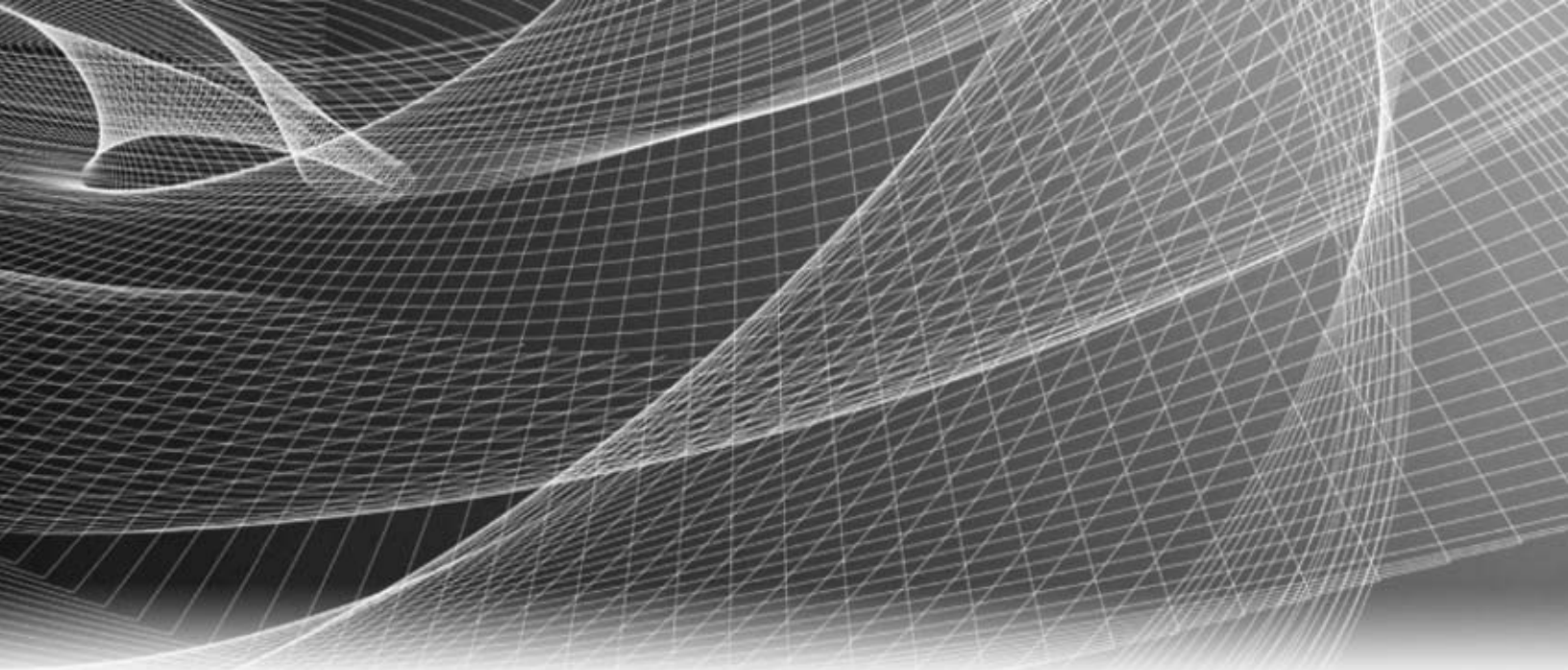




EMC[®] Watch4net[®]

Version 6.3

Alerting Guide

P/N 302-000-602

REV 01

Copyright ©2013 - 2014 EMC Corporation. All rights reserved. Published in the USA.

Published January 2014

EMC believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

The information in this publication is provided as is. EMC Corporation makes no representations or warranties of any kind with respect to the information in this publication, and specifically disclaims implied warranties of merchantability or fitness for a particular purpose. Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

EMC², EMC, and the EMC logo are registered trademarks or trademarks of EMC Corporation in the United States and other countries. All other trademarks used herein are the property of their respective owners.

For the most up-to-date regulatory document for your product line, go to the technical documentation and advisories section on the EMC online support website.

Chapter 1 Alerting	7
About alerting	8
How alerts are displayed	9
How APG parses report data	10
Alerting prerequisites	12
Configure the alerting engine	13
Configure alerting communication	14
Set up report alerts	15
Alert on a scheduled report	16
Immediately alert on a scheduled report	17
Access the Alerting page	18
Create and edit adapters	19
Create alert templates	21
Use preconfigured alert templates and examples	23
Set up a Log template	24
Set up a Trap Notify and Trap Clear template	25
Activate or deactivate alerts	26
Create alert definitions	27
About grouped boxes	29
Create a Grouped Box	30
About probing alerts	31
Prepare for probing	31
Probe an alert definition	32
Probe a report	34
Alert definition reference	36
Entry Point filter	36
Operations	37
Conditions	40
Actions	42
Index	45

Chapter 1 Alerting	7
To configure alerting communication	14
To alert on a scheduled report	16
To immediately alert on a scheduled report	17
To create or edit an adapter	19
To create an alert template	21
To access examples of alert definitions	23
To set up a Log template	24
To set up a Trap Notify or Trap Clear template	25
To activate or deactivate an alert definition	26
To create an alert definition	27
To create a Grouped Box	30
To test an alert definition	32

Alerting

Topics

- [About alerting](#)
 - [How alerts are displayed](#)
 - [How APG parses report data](#)
 - [Alerting prerequisites](#)
 - [Configure the alerting engine](#)
 - [Configure alerting communication](#)
 - [Set up report alerts](#)
 - [Alert on a scheduled report](#)
 - [Immediately alert on a scheduled report](#)
 - [Access the Alerting page](#)
 - [Create and edit adapters](#)
 - [Create alert templates](#)
 - [Use preconfigured alert templates and examples](#)
 - [Activate or deactivate alerts](#)
 - [Create alert definitions](#)
 - [About grouped boxes](#)
 - [Create a Grouped Box](#)
 - [About probing alerts](#)
 - [Prepare for probing](#)
 - [Probe an alert definition](#)
 - [Probe a report](#)
 - [Alert definition reference](#)
-

About alerting

You can create alerts for almost any kind of data parsed by APG: performance data, event data, report data and alert data from external sources that doesn't enter standard APG performance or event pipelines.

APG has three ways to create alerts

Templates enable you to pre-configure values and add them to an alert definition.	Create alert templates Use preconfigured alert templates and examples
Grouped boxes containers of pre-configured reusable linked components that give you complex alerting functionality. Any changes you make to a grouped box is reflected in all its instances.	About grouped boxes Create a Grouped Box
Alerting on reports You can set up alerts for reports, showing the results of formulas, baselines, and other aggregated data over time. This can give you more complete information than other types of real-time data sent to alerting such as raw timeseries and event data.	Alert on a scheduled report Immediately alert on a scheduled report

About APG alerting modules

The APG Alerting Backend and Frontend receive and alert on data. An instance of each is installed in a default APG installation.

Alerting Backend

The Alerting Backend receives data and alerts on it. You can have multiple Alerting Backends on different servers to distribute the load. Each Alerting Backend contains alerting configuration such as alert definitions, adapters, and other dependencies that are set up in the Alerting Frontend.

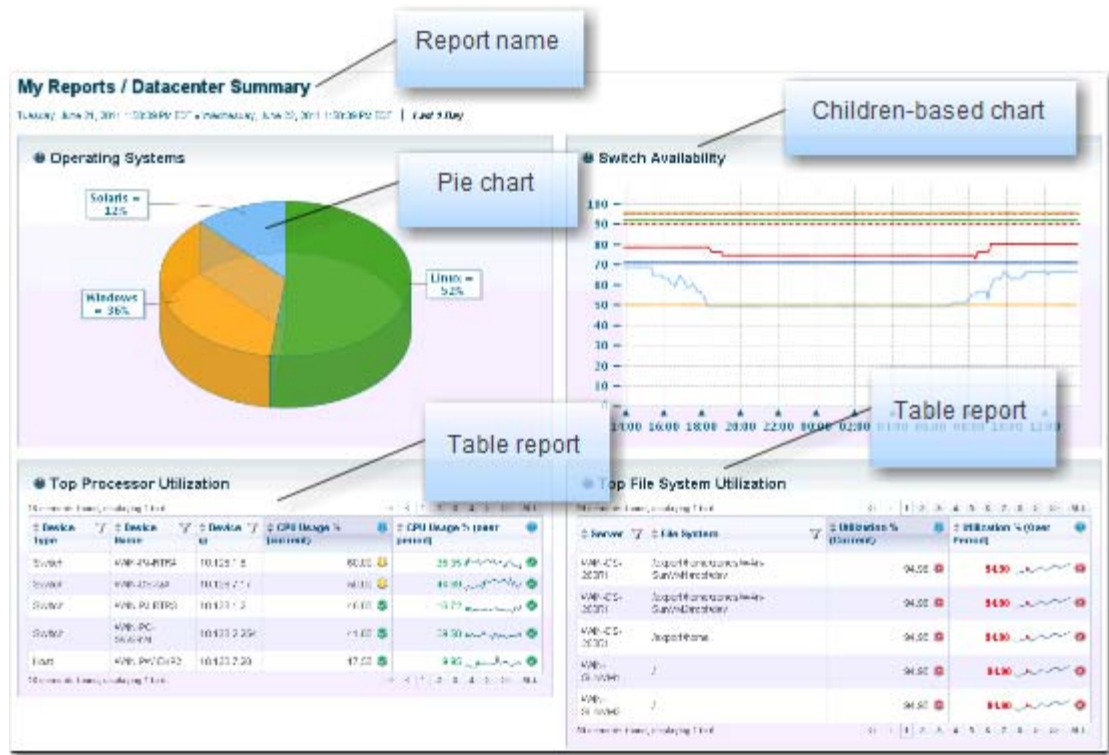
An Alerting Backend can receive data from APG or a third party. For example, it can receive performance data sent by a Collector Manager through a Cache Connector, and events and flows from an Event Processing Manager by a Generic Event Writer. It can also receive reports to alert on from the APG Frontend or a third party.

Alerting Frontend

The Alerting Frontend is where you define alert definitions and their dependencies. It interfaces with one or more Alerting Backends.

How alerts are displayed

This is an example of how alerts can be displayed in a Mixed Report.



Report name	The report name, Datacenter Summary, is used as the value for the reportName property for each line of alert data.
Pie chart	Alert data is generated for each operating system. The value for Linux is 52. This percentage represents its relative value compared to the other operating systems. Its absoluteValue is 13, which is the number of devices that use this operating system. Its totalValue is 25, which is the total number of devices computed in the report.
Children-based chart	Six lines of alert data appear for every timestamp for each line of the graph, which represents the availability of a switch on the network. The data is real-time, derived from a four minute polling interval, so over the course of a 24 hour period there are 360 timestamps. Multiplied by six metrics, the child report has 2160 lines of alert data, each of which is individually parsed by alert definitions. Each line of data contains a name property, whose value is populated by the legend entry, which in this case is the name of the device and the location. Although the legend is hidden in mixed reports, its values are used.
Table reports	The first Table report has two alerting data lines for each table row, as there are two value columns. Each row has an alerting data line with CPU Usage% (current) as its name and the value of the metric. This line also has Device Type, Device Name, and Device IP properties. The second Table report has similar results. For each row there's one alerting data line with Utilization% (Current) as its name and the value of the metric. This line also has Server and File System properties.

How APG parses report data

The APG Report Data Adapter parses the data in the reports it receives. The data and related properties depend on the report type. This section describes what data and properties are available from reports parsed by the APG Report Data Adapter so that you can create alert definitions and edit reports in the APG Frontend for alerting.

Each data point in a report, such as a point in a graph or a table row with a value column, results in a line of data that can be evaluated by an alert definition. Each line of alerting data output by the APG Report Data Adapter has a timestamp, value and name, in addition to other properties.

Properties

These properties are added to every data line from the APG Report Data Adapter:

- The name of the report is added as the `reportName` property.
- The displayed properties with their values from the header of the report are included with the lines of alerting data from the report. If it is a mixed report, the lines of alerting data from each child report have these common properties, although usually properties are used only in individual reports for a single device.

Tables

Each row of a Table report results in one line of alerting data per value column that can be evaluated by an alert definition according to these criteria:

- The value column of the table is used for alert definition evaluation, unless something else is specified for evaluation, such as a field in a string operation.
- The column name of the value column is used as the property value of the name property. The name is what the report editor named the column.
- Data from other table columns are input as additional properties, with their column names used as the property names. The name is what the report editor named the column.

Graphs

In a standard timeseries graph, each timestamp (point on the graph), generates a line of alert data that can be evaluated. In a graph with four lines (metrics), at each timestamp there are four lines of data, each evaluated by an alert definition.

The value of the property name is derived from the legend, even if the legend is hidden as it is in Mixed reports.

For Horizontal bar and Pie charts, each category in the graph results in a line of alerting data that contains:

- the relative value in percentage of the metric compared to the other categories
- the value of the name property, which is the name of the report
- the timestamp of each line of alerting data from these types of reports, which corresponds to when the report was generated rather than the time period the report covers. We recommend you don't use these type of reports for time-based alerting.
- the section property for each line of alerting data, which corresponds to the legend for the category reported on in the data
- the `absoluteValue` which is the absolute value of the metric for the category for the report period
- the `totalValue` which is the sum of the absolute values for every category in the graph.

Baseline graphs

Baseline graphs are like other timeseries graphs with these additional characteristics:

- One alerting data line is created for each timestamp with the value of the metric and its name, according to the contents of the legend, using the value of the name property.
- The `baselineValue` property is added, which is an average for the metric at the specified time over the last month.
- The `baselineMin` is added, which is the minimum value recorded for the metric at the specified time over the last month.
- The `baselineMax` is added, which is the maximum value recorded for the metric at the specified time over the last month.

You can use a formula to vary the length of the baseline instead of using the default, which is the past four weeks.

Alerting prerequisites

To alert on reports, you must set these up:

- the Alerting Frontend and the Alerting Backend.
See [Configure the alerting engine](#).
- communication between the APG Frontend and the Alerting Backend if they are on different servers.
See [Configure alerting communication](#).
- an APG Report Data Adapter in the Alerting Frontend. This is created automatically the first time a report is parsed by alerting. See [How APG parses report data](#).
- alert definitions in the Alerting Frontend to gather data from reports.

Configure the alerting engine

If you are an APG administrator, you must configure the service of the Alerting Backend to make alerts accessible to the Alerting Frontend. You configure the Alerting Engine with this set of JVM arguments.

JVM argument for Alerting Engine configuration

File	Mandatory	Description
alerting.config.file	✓	configuration file containing the list of definitions, adapters, components and templates
alerting.snmp.oid	✓	SNMP OID used by the SNMP trap action
alerting.max.crunching.thread		number of threads used for the crunching of incoming data. Default is 5.
alerting.conf.dir		directory containing the configuration schema files. Default is AlertingBackendDir/conf.
alerting.custom.conf.dir		The directory of the custom configuration files for some components. Default is AlertingBackendDir/custom.
alerting.suggestion.sampling		sampling of properties catching for the suggestions. Default is 100.
alerting.suggestion.limit		The number of values kept for each properties. Default is 1000.
alerting.mail.from		from address of a mail action. Default is noreply@mybusiness.com.
alerting.probe.size		probe buffer size. Default is 5MB.
alerting.probe.ttl		probe time to live in seconds. Default is 60 seconds.

Although it isn't mandatory, alerting will most of the time be used with the APG values. To enable collecting these values, edit the Alerting connector in the *Collecting/Collector-Manager/Default/conf/collecting.xml* file from the APG installation directory. The connector should look like this:

```
<!-- This collecting component connects the collecting process to the Alerting itself, using a
plain text socket. -->
<connector enabled="false" name="Alerting" conf="Collector-Manager/Default/Cache-Connector/conf
/alertingconnector.xml" />
```

The file *alertingconnector.xml* creates a socket on port 2010, which is the listening port of the default APG Values Socket Listener adapter of the Alerting Backend. By enabling both the connector and the adapter, Alerting is able to trigger on APG values.

Configure alerting communication

If the APG Frontend and the Alerting Frontend are not on the same server, add lines to the APG.xml file so that the Frontend knows which Alerting Backend instances it can send reports to.

This procedure assumes you have set up your APG Frontend and Alerting Frontend and Backend.

To configure alerting communication

- 1 On the server where the Alerting Frontend resides, open **alerting-frontend.xml**, located at <APG>/Web-Servers/Tomcat/<Instance>/conf/Catalina/localhost.
- 2 Copy the alerting properties.
For example, if you have two Alerting Backends configured, you copy these lines:

```
<Resource name="manager/Local Manager" auth="Container"
type="com.watch4net.alerting.jmx.AlertManagerFactory"
factory="org.apache.naming.factory.BeanFactory" user="admin" pass="changeme" url="service:jmx:rmi://jndi/rmi://localhost:52569/jmxrmi" />
<Resource name="manager/Second Manager" auth="Container"
type="com.watch4net.alerting.jmx.AlertManagerFactory"
factory="org.apache.naming.factory.BeanFactory" user="admin" pass="changeme" url="service:jmx:rmi://jndi/rmi://localhost:52570/jmxrmi" />
```
- 3 On the server where the Alerting Frontend resides, open **APG.xml**, located at <APG>/Web-Servers/Tomcat/<Instance>/conf/Catalina/localhost and paste the lines you copied from the alerting-frontend.xml file anywhere in the <context> tags.
- 4 If the Alerting Frontend references the local Alerting Backend, replace localhost in each copied entry with the IP address of the Alerting Backend.
- 5 Restart Tomcat.

Set up report alerts

You can alert on reports, taking into account results of formulas, baselines, and other aggregated data over time. This can give you more complete information than other types of data sent to alerting, such as raw timeseries or event data, which are usually real-time and do not include the enrichment and organization provided by the APG reporting engine.

If you are using a baseline report, you can use a formula to vary the length of the baseline instead of using the default of the past four weeks.

There are two ways to send reports to be alerted upon:

- By indicating which Alerting Backend to send a scheduled report to.
[Alert on a scheduled report.](#)
- By immediately sending a scheduled report to the Alerting Backend.
[Immediately alert on a scheduled report.](#)

Alert on a scheduled report

When you alert on a scheduled report, you can send it to one or more Alerting Backends. This triggers alerts each time the scheduled report is generated. Reports should be sent to the Alerting Managers where the alert definitions that act on the report data are set up. Contact your administrator for this information. Sending data to an alerting backend doesn't necessarily trigger any actions as most of the time the state of the analyzed components does not meet any alerting definitions.

To immediately generate a report to be alerted on see [Immediately alert on a scheduled report](#).

To alert on a scheduled report

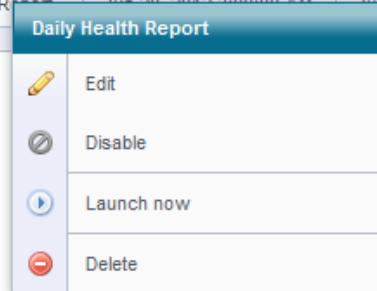
- 1 In the **APG Frontend**, in the tree click **Scheduled Reports**.
- 2 Click the report you want to alert on and click **Edit**.

Scheduled Reports

This list contains all reports that have been scheduled, as well as some statistics about their generation. Scheduled reports will be generated at the given rate, and then sent by email once finished.

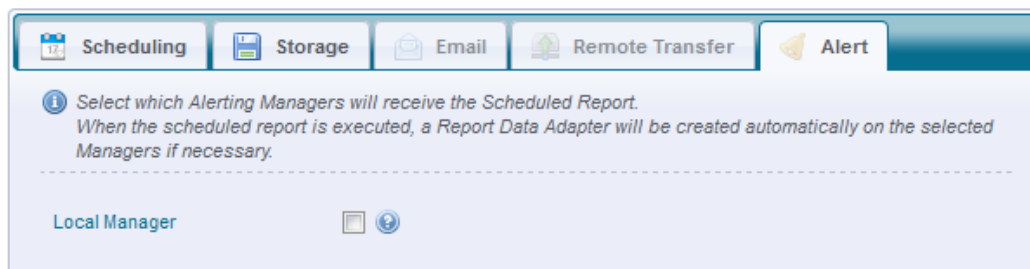
Click on a line to access the corresponding scheduled report information.

Name	Next Run	Last Run	Last Duration	Status
Daily Health Report	Jun 20, 2012 3:00:00 AM	Jun 19, 2012 3:00:00 AM	17 secs	



- 3 Click the **Alert** tab.

Editing Scheduled Report



- 4 From the list of Alerting Managers, select where you want the report to be sent and click **Save**.

Immediately alert on a scheduled report

You can generate a scheduled report whenever necessary, for example to trigger an alert on it on demand, or to test it with probes in the Alerting Frontend.

To immediately alert on a scheduled report

- 1 In the APG Frontend, in the tree click **Scheduled Reports**.
- 2 Click the report you want to generate and click **Launch now**.

Scheduled Reports

This list contains all reports that have been scheduled, as well as some statistics about their generation. Scheduled reports will be generated at the given rate, and then sent by email once finished.

Click on a line to access the corresponding scheduled report information.

Name	Next Run	Last Run	Last Duration	Status
Daily Health Report		Jun 19, 2012 3:00:00 AM	17 secs	✓

Daily Health Report

 Edit

 Disable

 **Launch now**

 Delete

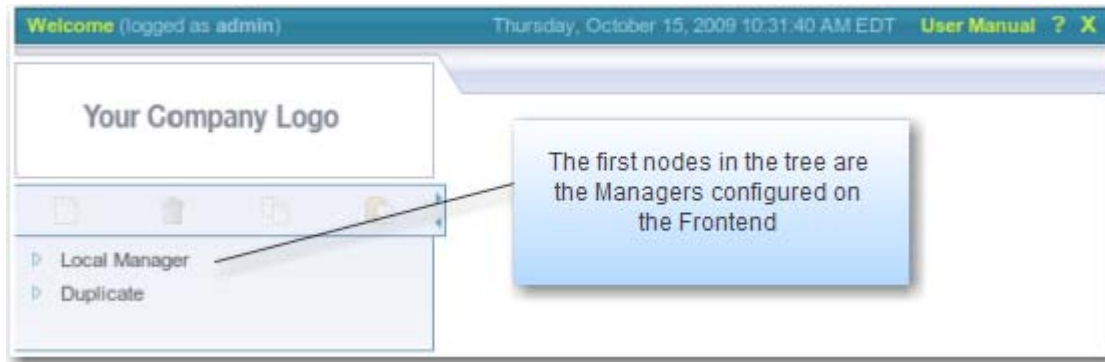
Launch the scheduled report now.

Access the Alerting page

To access the Alerting page, go to the address of your Tomcat server alerting frontend. This is usually `http://localhost:58080/alerting-frontend`.

or

in APG click **Administration**, and in the **Configuration** area click **Alerting**.



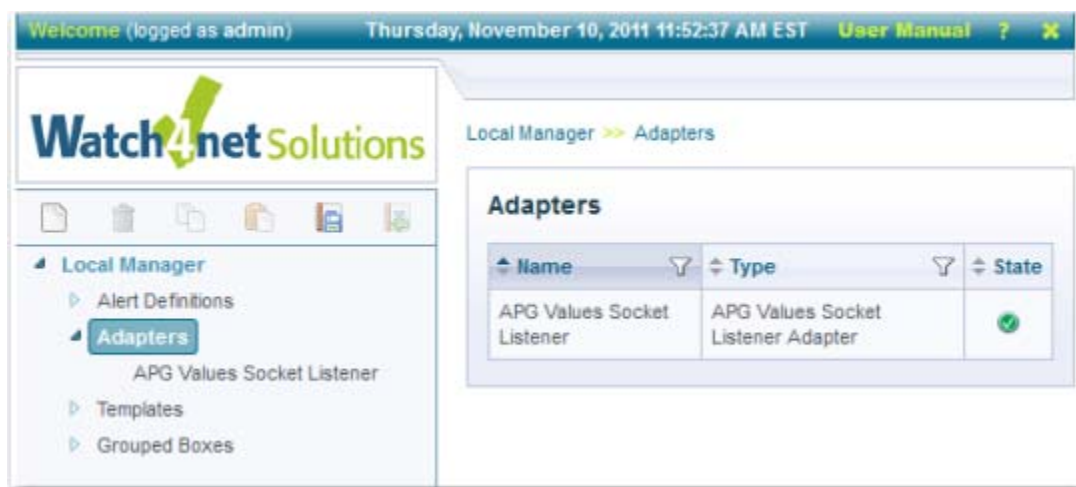
Tools

The tools above the report tree can be unavailable depending on the node you select.

	Creates a new element under a list node and opens the element edit page. If you select another node before saving the new one, the new element is deleted.
	Deletes the element.
	Copies the element.
	Pastes the copied node if it is a list that contains the same type of elements.
	Exports the element to an XML file.
	Imports an XML file to the Manager.

Create and edit adapters

An adapter takes data and translates it to alerting data.



To create or edit an adapter

- 1 In the tree click **Adapters** and above the tree click the **New element** icon.



New Adapter

*This adapter will listen on a socket for incoming standard APG Raw Values. The resulting data will have the properties **adapterName** containing the *name of this adapter* and **adapterType** containing *RVSockeListener*.*

Type: APG Values Socket Listener Adapter ▼

Name: APG Values Socket Listener Adapter

Parameters:

Port:

- 2 From the Type list select the type of adapter you want to create.

Values Socket Listener Adapter

This adapter waits for values from the socket connector in the collecting.xml file. The data has the properties **adapterName** containing the name of this adapter and **adapterType** containing **RVSockeListener**.

- In the **Name** box type a descriptive name for the adapter.
- In the **Port** box type the port that the adapter will take its incoming raw data from.

Events Adapter

This adapter waits for events from a generic-event-writer in the processing.xml file. The data has the properties adapterName containing the name of this adapter and four parameters.

- In the **Name** box type a descriptive name for the adapter.
- In the **Value** field you can optionally enter the name of the field containing the decimal value of the event to be used in the operation, such as bytes for flow events. You can leave this box blank for a strings only event.
- In the **Properties** field enter a comma-separated list of the properties used in the entry point or special components. If the default data type of a property does not give the best display format, enter the data type of the field with the field name separated by a comma, for example SRC_PORT:INT. The data types can be Boolean, byte, bytes, double, float, int, long, numeric, short, object, or string. The data type is implementation-specific for the type of data the adapter is parsing
- In the **Timestamps** field you can optionally enter the field that contains the timestamps of the events. Otherwise, the current timestamps are used.
- In the **Port** box type the port that the adapter will take its incoming raw data from.

Report Data Adapter

This adapter parses exported APG reports. The properties of the data are the report displayed properties and the report name (property = reportName). Table reports contain the non value columns and the name of the value column (property = name). Graph reports have the legend of each series (property = name). Each file parsed is deleted after parsing. The resulting data has the properties adapterName containing the name of this adapter and adapterType containing APGXmlReportAdapter.

- In the **Name** box type a descriptive name for the adapter.
- In the **Time check** box enter the time in minutes between each check.
- In the **Directory** box you can optionally enter the subdirectory under the Alerting custom directory that contains the files to parse.
- In the **File name regex** box enter the file name regex. You can use the * wildcard character.

File Data Adapter

This adapter parses each file corresponding to a file name regex in a directory. Each file parsed is deleted after parsing.

- In the **Name** box type a descriptive name for the adapter.
- In the **Directory** box you can optionally enter the subdirectory under the Alerting custom directory that contains the files to parse.
- In the **Time check** box enter the time in minutes between each check.
- In the **File name regex** box enter the file name regex. You can use the * wildcard character.
- Upload any parser or formatter files you want to use.

Create alert templates

Alert Manager comes with preconfigured templates. You can edit the existing templates or create new ones. See [Use preconfigured alert templates and examples](#).



To create an alert template

- 1 In the tree click **Templates** and above the tree click the **New element** icon.



New Template

i This operation will filter the flow of incoming data for this definition. Remember that the wizard will keep only a samples of the properties so if a property is missing, its not necessarily unknown.

Family
Entry Point ▼

Group
Filtered Entry Point

Type
Filtered Operation

Name

Description

Filter

- 2 From the **Family** list select the type of template you want to create:

Type	Description	For details see
Entry Point	The Entry Point filter works like report filters on the APG Frontend. It filters the flow of incoming data for alert definition.	Entry Point filter
Operation	Operations evaluate incoming data to see if they should go to the next element or be dropped.	Operations
Condition	A condition verifies a condition of the data before sending it to its actions or operations. Conditions can be stateful or stateless. In a stateless condition, new data is sent to the next component. A stateful condition sends the data only when the state is changed. For example, if state of ID A = true, it sends data only when the state is false.	Conditions
Action	Actions inform users of new or resolved problems in their system or run an auto resolving process.	Actions

- 3 Enter a name and description for the template.
- 4 Use the Filter Wizard to select the properties you want to filter on.
- 5 Click **Save**.

Use preconfigured alert templates and examples

Alert Manager comes with preconfigured templates and examples. You can edit the existing ones or create new ones based on the templates and examples.

To access examples of alert definitions

- 1 In the tree click **Alert Definitions** > **Examples**.

Log and Traps are the most frequently used actions in alert definitions. APG has a log template and two trap templates that use the SNMP trap configuration.

- [Set up a Log template](#)
- [Set up a Trap Notify and Trap Clear template](#)
- [Activate or deactivate alerts](#)

Set up a Log template

This example shows creating a new template based on the Log template. Logs are frequently used actions in alert definitions.

To set up a Log template

- 1 In the tree click **Templates > Log Template**.

Log Template

This action will write an entry in a log file.

Family	Action
Group	Log
Type	Log
Name	Log Template
Description	

Parameters

File prefix	Alerts/log
Logs retention	5
Rotation time	1440
Rotation entries	5000
Entry content	PROP:'device' with PROP:'part' -> PROP:'name' = VALUE

- 2 In the **Description** box enter a description for the new template.
- 3 In the **File prefix** box enter the directory path and prefix for each log file. The log files are created in the Alerting-Backend/instance/Alerts folder and named log_INDEX.log, where index is incremented until it reaches the log retention limit. The newest log files have the lowest index, and the current log is the 0 log.
- 4 In the **Log retention** box enter the number for how many logs you want to keep.
- 5 In the **Rotation time** box enter a value for the maximum time in minutes that log data is gathered before the log is created. This value is optional if you are setting a value for rotation entries. In this example, the log files are rotated once a day (1440 minutes).
- 6 In the **Rotation entries** box enter a value for the maximum number of entries per log file. This value is optional if you are setting a rotation time. In this example, the log is rotated when the 0 log file has 5000 entries.
- 7 In the **Entry content** box enter the properties for the log entry. To add the value, date or ID of the current data, use the keywords VALUE, TMST or ID. To add a property of the current data, use the keyword PROP:'propertyName'. In this example, each line of the log contains: DEVICE with PART -> NAME = VALUE.
- 8 Click **Save**.

Set up a Trap Notify and Trap Clear template

This example shows creating a new template based on the Trap Notify template. Traps are frequently used actions in alert definitions. APG has a log template and two trap templates that use the SNMP trap configuration. A Trap Notify alert sends an alert to your fault management system, such as EMC Ionix. The Trap Clear deactivates the alert.

To set up a Trap Notify or Trap Clear template

- 1 In the tree click **Templates > Trap Notify template**.

Trap Notify Template

This action will send an SNMP trap when activated.

Family	Action
Group	SNMP Trap
Type	SNMP v1 Trap
Name	Trap Notify Template
Description	

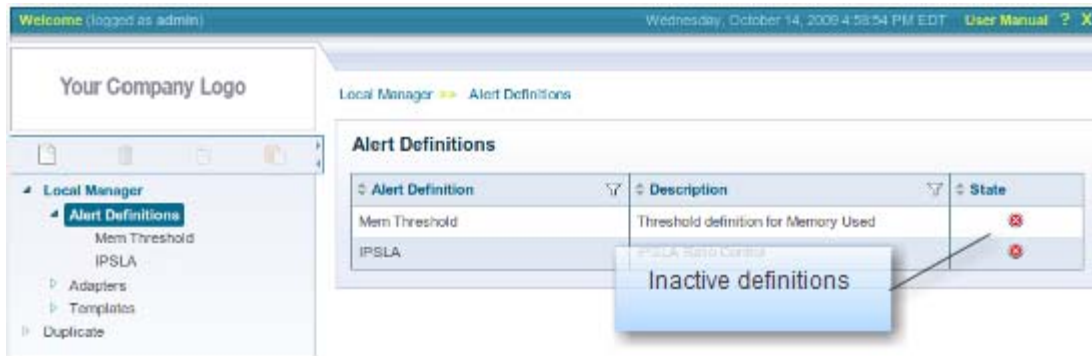
Parameters

Host	127.0.0.1
Port	162
Community	public
Generic ID	6
Specific ID	1
Trap content	VALUE,PROP:'ADDED_THRESHOLD',2,PROP:'device',PROP:'parttype',PROP:'part',PROP:'unit',This is an alert from APG,'userdef1','userdef2','userdef3'

- 2 In the **Description** box enter a description for the new template.
- 3 In the **Host** box type the host that will receive the traps. The host is the address of the SNMP trap manager and should be changed if it's not on the same host as the alerting backend.
- 4 In the **Port** box type the trap manager port. 162 is the default for this protocol.
- 5 In the **Community** box type the trap community. Public is the default.
- 6 In the **Generic ID** box type the generic ID of the trap.
- 7 In the **Specific ID** box type the specific ID of the trap format for the generic type.
- 8 In the **Trap content** box type the content for each variable of the trap, separated by commas. Commas will be escaped if they are preceded by a \. To add the value, date or ID of the current data, use the keywords VALUE, TMST or ID. To add a property of the current data, use the keyword PROP:'propertyName'. We recommend you do not change the content of the traps as the order of the properties match our trap configuration. You can change the last three properties to ones you define. The only difference between Trap Notify and Trap Clear is the number value in the Trap content: 2 is notify and 4 is clear.

Activate or deactivate alerts

You can make definitions active or inactive.



To activate or deactivate an alert definition

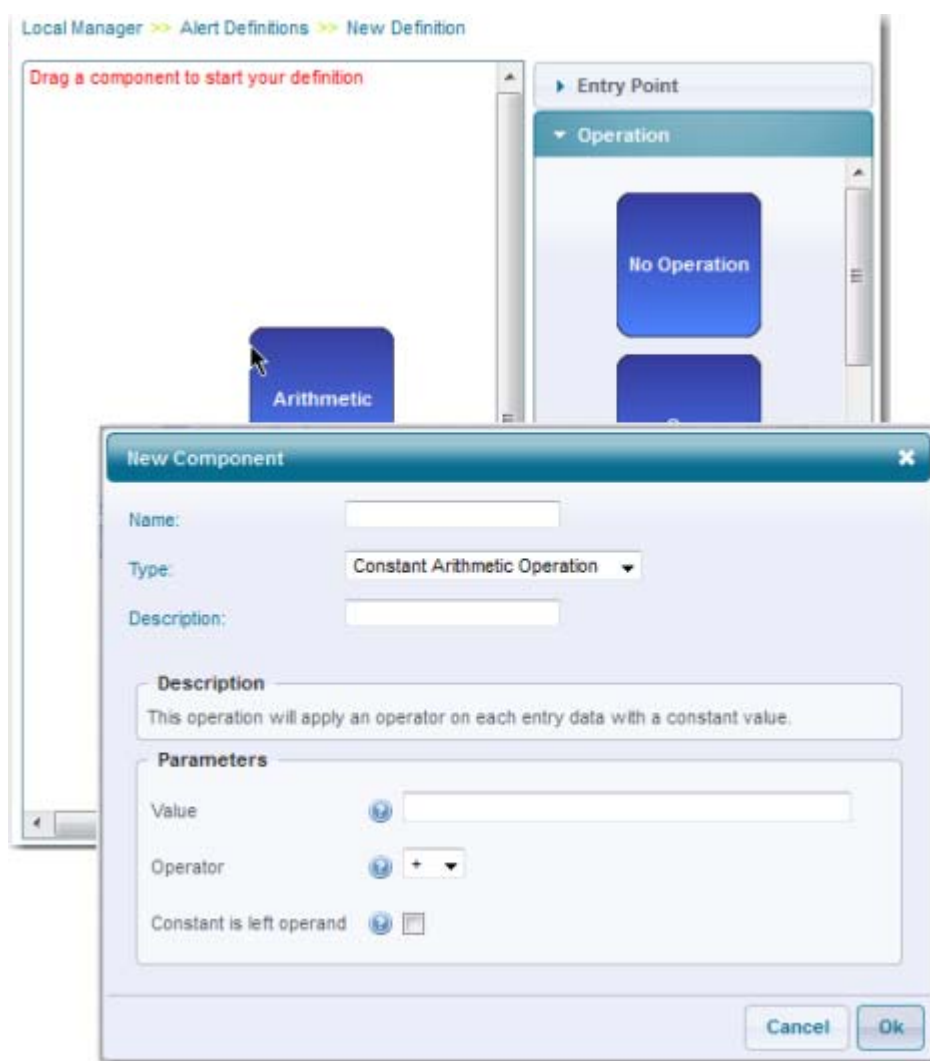
- 1 In the tree click **Alert Definitions** and click the alert you want to activate or deactivate.
- 2 Choose **Enable** or **Disable**.

Create alert definitions

You can view alert definition examples to help you get started with constructing alert definitions. In the tree click **Alert Definitions > Examples**.

To create an alert definition

- 1 In the tree click **Local Manager > Alert Definitions** and above the tree click the **Create a new element** icon. To edit a definition, select it in the tree.
- 2 Drag the **Filtered Entry Point** onto the whiteboard and define it. See [Entry Point filter](#) for how to set up the entry point.
- 3 Expand the type of component you want to work with, and drag it onto the whiteboard. This example shows adding an arithmetic component.

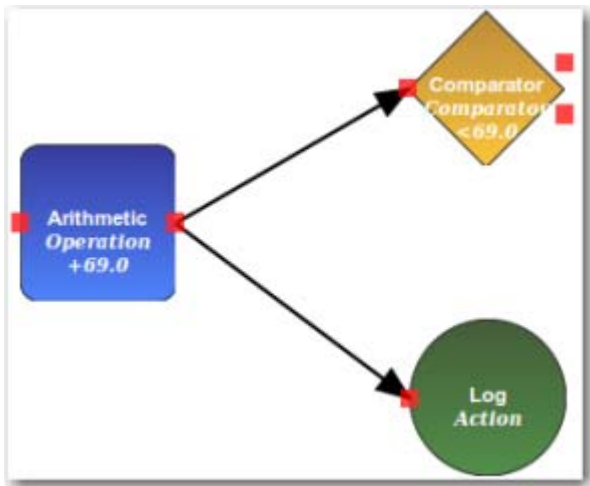


4 For how to define components, see the [Alert definition reference](#):

- [Operations](#)
- [Conditions](#)
- [Actions](#)



Every component has connectors, which are red points. To link components, click a connector on the first component, and then the connector on the component you want to link to. A connection arrow appears, linking the components.



There is no limit to the number of objects you can link to the same output.

You can move components by dragging them.

The screenshot shows a 'New Definition' dialog box with a title bar and a close button. It contains three input fields: 'Name:', 'Description:', and 'Folder:'. The 'Folder:' field has a dropdown menu showing 'Alert Definitions'. At the bottom, there are three buttons: 'Cancel', 'Save and disable', and 'Save and enable'.

When you are finished creating the alert definition, click **Save**.

Enter a name and description for the alert, and click **Save and enable** to apply the alert definition immediately, or **Save and disable** if you want to enable the definition later.

To enable it afterwards, in the tree click **Alert Definitions**. On the **Alert Definitions** page click the alert you want to enable > **Enable**.

About grouped boxes

With grouped boxes you can:

- Create a subset of boxes such as baseline computations that you can reuse in multiple definitions. Grouped Boxes can contain any alert definition component and can easily be reused.
- Create a single box that has its instances updated whenever it's modified, unlike templates that don't have a link. When you make a change to a grouped box, the change is immediately propagated to all the alert definitions that use it.

Grouped boxes are containers of standard alerting components, linked in the same way as alert definitions, that you can use in an alert definition or other grouped boxes.

Unlike alert definitions, grouped boxes have free inputs and outputs on any number of components, whether they're linked or not. The free connectors appear as inputs or outputs on the group box that you can use to link to other components. You can mouse over a free connector to see its source component, which is its name, and the input/output type, for example Condition met output.

Grouped boxes are created like alert definitions with these exceptions:

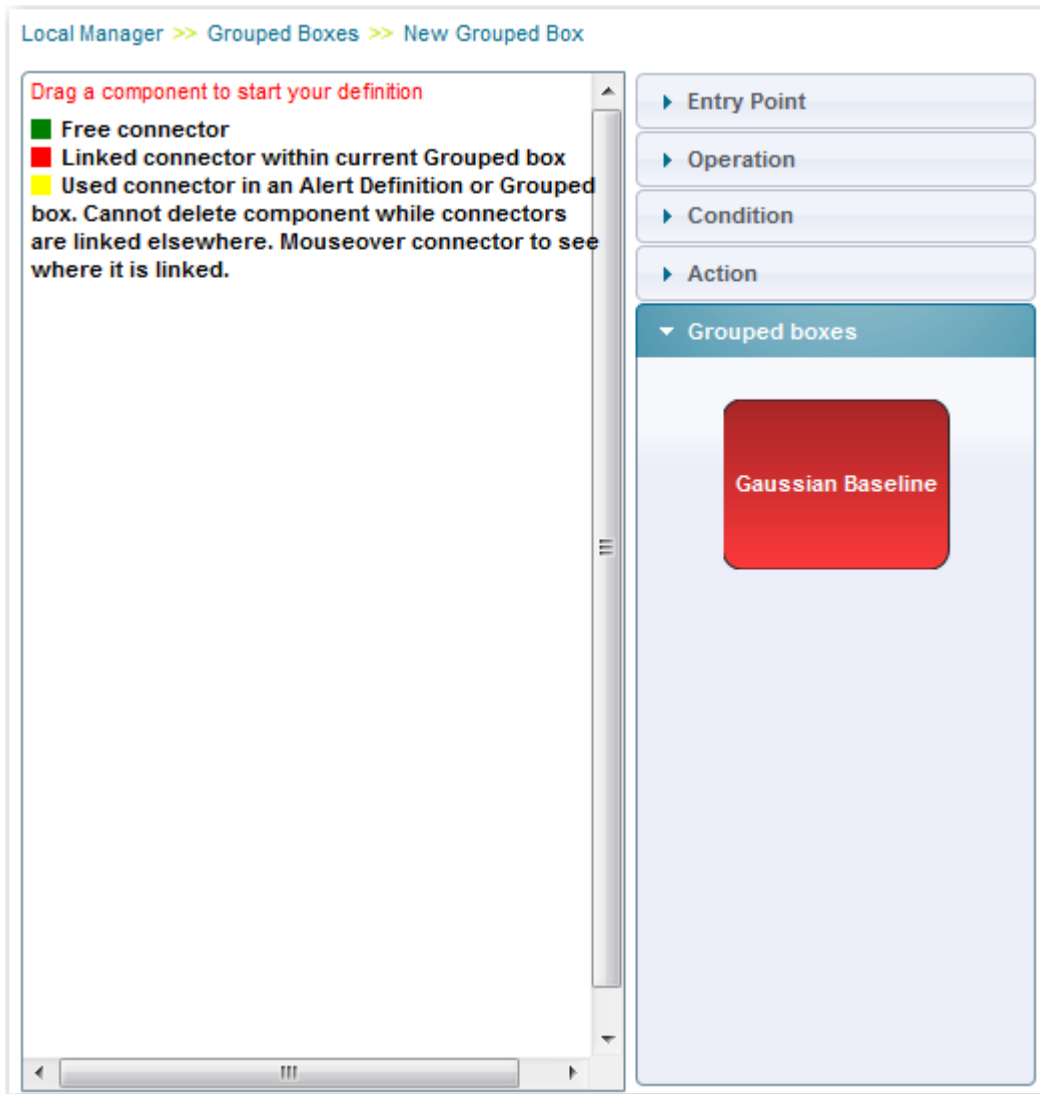
- Entry points aren't required to save a grouped box.
- Connectors have three possible states: red, yellow and green.
- When you save a grouped box, the resulting box in the definitions is a red box and its connectors are the unused input/output connectors of its internal components. No configuration is needed when using a grouped box in a definition. The description of the configuration is the description you enter when you save the grouped box.

Create a Grouped Box

You can create and edit Grouped Boxes and use them in alert definitions. Grouped Boxes can contain any alert definition component and can be easily reused.

To create a Grouped Box

- 1 In the tree click **Local Manager > Grouped Boxes** and above the tree click the **Create a new element** icon. To edit a Grouped Box, click it in the tree.



- 2 Edit the Grouped Box definition by dragging entry points, operations, conditions and actions on the page, and editing their parameters. Unlike alert definitions, entry points are not mandatory for grouped boxes. For how to set up the parameters see the [Alert definition reference](#):
 - [Entry Point filter](#)
 - [Operations](#)
 - [Conditions](#)
 - [Actions](#)

- 3 Link the components in a logical flow for how you want the data to be evaluated. Connectors have three states:
 - **Green** represents free input or output connectors. You can link within the scope of the Grouped Box you are working on or leave the connector free, in which case it appears as free a connector on the Grouped Box when it is used in an Alert Definition.
 - **Red** represents a connector that is linked with another element within the Grouped Box. The data flow between these connectors takes place in the Grouped Box, and the connectors are not visible and cannot be linked to in the Grouped Box.
 - **Yellow** represents connectors that have been used within an Alert Definition. You can mouse over a yellow connector and see a list of the alert definitions that use it, including the component that links to it. A yellow connector cannot be linked to in the source Grouped Box as it is already occupied in one or more Alert Definitions, and it cannot be deleted until it is freed in the Alert Definitions it is linked to.
 - You can use a Grouped Box within another Grouped Box but not within itself. Grouped Boxes used within a Grouped Box display their available connectors.
- 4 Click the **Save** button.

About probing alerts

Probing helps you confirm the behavior of an alert definition. Do a probe to test the elements in an alert definition in real-time to see how data is evaluated at each component, and how many times actions have been triggered.

Probes help you spot problems with how you construct your alert definitions, so that you can adjust them and retest until your alerts are satisfactory. This helps you to construct ever more complex alerts to match your network's increasing complexity.

Probing is available only to users who have write permission.

Prepare for probing

Before you start, you can make these changes to improve the success of probing your alert definitions:

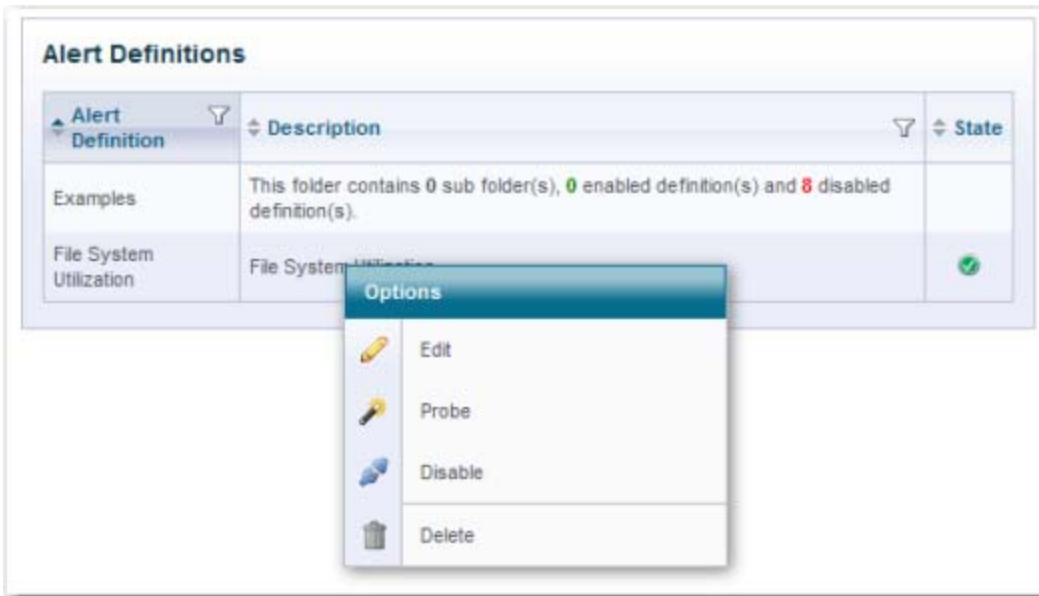
- Make sure the alerting sources, such as timeseries, events, text, or report data, are properly configured to be sent to the Alerting Backend.
- Reduce the polling intervals of one or more collectors, or how often source data is sent to an Alerting Backend, so that your probing yields more results that correspond to the time frame. If, for example, the default polling for the Smarts Collector uses 4 minutes, you would have to wait 4 minutes for every probe result. You can change the Smarts Collector to 10 seconds for the duration of the probing, and then switch it back for production.
- Reduce the sampling property that determines how many properties per data sample are preserved. This is controlled by the *alerting.suggestion.sample* property in the *module.properties* file located at `<APG>/Backends/Alerting-Backend/<instance>/conf`. By default this property does not appear, and 1 in 100 properties are kept. For example if polling takes place every ten seconds, it can take a long time to accumulate enough properties for effective probing. You can change this to 1:1 for the duration of your probing session.
- Filter hints and sampled properties are dependent on the data coming into the Alerting Backend. You can increase the number of hints and sampled properties by adding and adjusting the *alerting.suggestion.sampling* property in the *module.properties* file located in the conf directory of the Alerting Backend.

Probe an alert definition

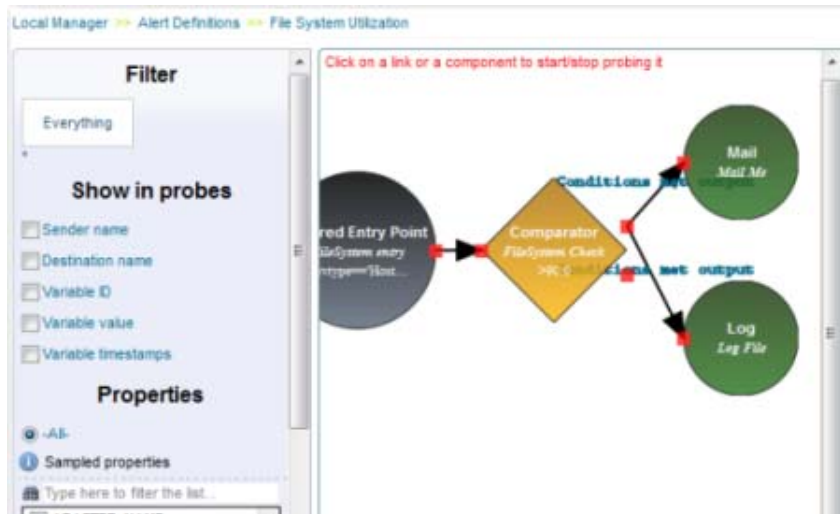
You can use probing to test an alert definition in real-time. Before you start, see [Prepare for probing](#) to set up your environment for the best probing results.

To test an alert definition

- 1 Enable or disable the relevant adapters:
 - In the tree click **Adapters**.
 - Click the adapter you want to use or disable and click Enable or Disable.
- 2 Enable relevant alert definitions:
 - In the tree click **Alert Definitions**.
 - Click the alert definition you want to test and click **Enable**.
- 3 In the **Alert Definition** list, click the alert definition you want to probe, and click **Probe**.



- 4 The alert definition whiteboard appears.



Filter	Applies a filter to the probe results. This affects only the probe results and not the alert definition. Use a standard APG filter. Only a subset of properties may appear in the wizard depending on how many samples were cached. You can adjust this by changing the value of the <i>alerting.suggestion.sampling</i> property, in the <i>module.properties</i> file, at <i><APG>/Backends/Alerting Backend/<instance>/conf</i> .
Show in probes	Selects the property to show in all probe results. The properties appear at the beginning of each result line followed by the selected sampled properties. They appear only when you probe a link. • Sender name: component the line of alerting data originated from. • Destination name: component the line of alerting data is flowing to. • Variable ID: report node the line of alerting data originated from. The variable ID format depends on the adapter type. • Variable value: value from the line of alerting data to be evaluated by the alert definitions, unless a string in the data is selected for evaluation, such as a string comparator. • Variable timestamps: timestamp associated with the line of alerting data. It is converted from a Unix timestamp to a human-readable form for clarity in the probes. For category graphs this is when the report was generated in the APG Frontend.
Properties	Selects properties whose names and values will appear in probe results. You can select All to display all the available properties.

- 5 Test or clear a component or link by clicking it. The component is highlighted and its information appears in an area of the same color.
- 6 To disable the Probe UI, for example to edit the definition to refine the probe results, click anywhere in the tree, and then click on the alert definition you were working with. From here you can edit it.

Probe a report

You can test how a report is alerted on. This example shares some guidelines for doing that. This procedure has three major steps:

- [Set up a scheduled report to be alerted on and generate it](#)
- [Create an alerting definition for the report](#)
- [Probe the alert definition](#)

Notes about report probe results

- Some components, such entry points and operations, may be less interesting to probe, as they return only their descriptions and settings.
- Actions, besides listing all their settings, show the number of times they have been executed. You can see in real-time the effect the report data had on the number of times the action executed by watching this number during the probing session.
- Enabling links displays the information in each line of alert data according to the selections in the Probe window. Since there is no filter all data is parsed and every property with its value appears. Depending on where the link is located in the alert definition, a subset of alerting data lines appear according to how they are routed from previous components in the chain.

Set up a scheduled report to be alerted on and generate it

- 1 In the **APG Frontend** go to **Global Reports > Dashboards > Datacenter Summary > Top File System Utilization**. This table has two value columns with lines of alerting data that can be evaluated by alert definitions. The Utilization% (Over Period) column shows the average file system utilization over the last day through the performance metric UtilizationPct.
- 2 To alert on file system utilization over the last month, change the reporting period. On the **Display** menu, in the **Time Selection** area, from **Over** select **1 month** from the second list and click **Apply**. This does not permanently change the report time range in the APG Frontend.
- 3 On the **Tools** menu select **Schedule/Schedule this Report**.
- 4 From the **Schedule** list select **Every month** to produce alerts monthly for the last month's file system use.
- 5 Click the **Alert** tab and select the Alerting Backend that contains the alert definitions to test with the data from this scheduled report. Alerting Backends are available if set up.
- 6 Now that the report is scheduled, to send it on demand to its Alerting Backend instances on the report tree click **Scheduled Reports**, click the report name on the **Scheduled Reports Management Page > Launch now**.
- 7 The APG Report Data Adapter, which parses reports from the Frontend, is automatically created the first time it receives a report from the APG Frontend to be alerted on. The APG Report Data Adapter is named `AUTO_CREATED_APG_REPORT_ADAPTER`. On the **Alerting** interface, change its **Time check** parameter to **1** (minute), so that reports received from the Frontend are parsed, and can be probed. You can change this back to a larger interval after the alert definitions are ready for production, to reduce the load on the server caused by needless polling.

Create an alerting definition for the report

- 8 On the **Alerting Frontend**, click **Alert Definitions > Examples > Multiple Severities Comparator** and above the tree click the **Copy** and then the **Paste** icon. Rename it **Utilization Over Month Alert**.
- 9 Click the newly created alert definition in the tree to display its components.
- 10 Mouse over its entry point and click the **pen** icon to edit it.
- 11 In the **Edit Components** dialog that appears, delete the contents of the filter. You can change its name and description to reflect its use of filtering on `Utilization%(OverPeriod)`.

- 12 You can add a filter later, but first test the unfiltered content using a probe, and include a filter from the Probe UI for iterative testing purposes. Alert definitions pass their data to the next component only if the state data has changed from a previous data line in a series. This is useful, for example, if instead of using the APG Report Data Adapter, you use the APG Values Socket Listener Adapter to directly input UtilizationPct metric data about the file systems received by the Smarts-Collector through the Collector-Manager, and do not want to continually trigger an alert when a file system is over a certain value, but only the first time it crosses the threshold, and subsequently when it is under the threshold. In this example the report will be parsed once a month and all the data will be parsed to see which thresholds are crossed for each line of alerting data. Therefore, edit each **Comparator** condition to enable their **Stateless** check boxes.
- 13 Save and enable the alert definition.
- 14 To make sure the APG Report Data Adapter named AUTO_CREATED_APG_REPORT_ADAPTER is enabled, click it in the adapter list and click **Enable**. You can disable any other adapters so that you can concentrate on the report data being received from the APG Frontend, if this does not impact other required running alerts.

Probe the alert definition

- 15 On the **Alerting** interface, in the tree click **Alert Definitions**.
- 16 Click **Utilization Over Month Alert > Probe**. The alert definition appears with the Probe window on the left, and space at the bottom of the screen for the probe results.
- 17 Enable a few of the components and links of the alert definition to see the data that will be flowing through them by clicking each one. For each component/link clicked, a probe result window appears at the bottom of the screen.
- 18 Open up another browser window, return to the **APG Frontend**, and on the **Scheduled Reports Management** page click the scheduled report you configured to be alerted upon > **Launch now**. The report generates and the results appear in the probe results.

Alert definition reference

- [Entry Point filter](#)
- [Operations](#)
- [Conditions](#)
- [Actions](#)

Entry Point filter

The Entry Point filter works like report filters on the APG Frontend.



This entry point inputs data from an SAA Probe that contains the JitterStatsNumOfRTT variable.

The difference between using a filter in the Alerting Frontend and the Backend is the suggestions shown in the wizard. The Alerting Backend is not bound to a database, so the list of known properties and values is created at runtime. Because of this, it's normal to have no suggestions when you use the wizard immediately after starting the Alerting Backend.

Operations

Operations evaluate incoming data to see if they should go to the next component or be dropped.

No Operation

This operation does not do anything. Use it if an entry needs separate senders in its process when all the entries come from the same operation, such as a Previous Polling Operation. It's a good way to make a used connector (red) of a grouped box available as an output if needed.

Case

This operation creates logical output groups that send data to each next operation until one has its condition met, such as a group of alert levels that send an alert only once for the highest level of alert. The order of criticality of the next components is the order of link creation, so the next component link is the highest level and if its condition is met when the next data is received, the other next components won't receive data.

Arithmetic

Constant Arithmetic

This operation applies an operator on each entry data with a constant value.

Value	value used by the operator
Operator	the arithmetic operator (+, -, / or *)
Constant is left operand	select this if the constant value must be the left operand

Data Arithmetic

This operation applies an operator on two data lines defined by the filter.

Operator	the arithmetic operator (+, -, / or *)
Time limit	maximum delta between the two values in minutes
Left operand	optional and needed only if you use an operator that produces a different result depending on the order of the operand, such as - and /. This parameter must contain the name of the sender who will send the data that must be the left side of the operation.
Group by	properties, each separated by a white space, that create a logical group of values for the operation
Value filter	optional and is necessary only if the entries of this operation come from the same input, as it defines two different elements of the same group

Math

Constant Math

This operation applies a complex mathematical operation to the data.

abs	the abs function which returns the absolute number
ceil	the ceil function which returns the smallest value but greater than the argument
exp	the exp function which returns the exponential value raised to the power of a double value
floor	the floor function which returns the largest value but less than the argument

Configurable Math

This operation applies a complex mathematical operation using a parameter to the data.

Operation	the mathematical operation
Value	value applied to the second parameter of the selected operation

Time Window

History Operation

This sliding window operation collects a range of values bounded by the number of values or the time range of the values. The output of the collected values can be their minimum, maximum, sum or average.

Wait for period	optional and for a time only bounded buffer, this flag is set if the buffer must wait a full period before computing values
Time range	optional and is the time window of the collected values in minutes
Window length	optional and is the values number to collect in the sliding window
Output	min, max, avg or sum

Previous Polling Operation

This sliding window operation keeps a series of previous polling for each data ID and distributes it to the outputs.

Time range	optional and is the time window of the collected values in minutes
Window length	number of previous polling to keep

Data Repeater

This operation blocks or allows output for every data that have their timestamps outside the time frame.

Update timestamps	select this to have the timestamps of each repetition be the current timestamp don't select this to have the actual timestamp used
Repetitions	repetition number for each ID. 0 = unlimited
Repeat period	time in minutes between repetitions

Date

This operation blocks or allows the output of data that doesn't have a timestamp within the time frame.

Block if in range	select this to block data if the timestamp is in the range you set here
Start time	inclusive start time of a day, following a hh:mm 24 hour format
End time	exclusive end time of a day, following a hh:mm 24 hour format. End time must be later than start time
Start day	optional. select this if the time range can be repeated for successive days, such as 8:00 to 17:00 Monday to Friday.
End day	

Data Creator

Properties To Values Creator

This operation takes a list of properties and creates a new value for each property that contains a numerical value. The original data is not sent to the next components. Enter a list with each property separated by a white space.

Timestamp To Value Creator

This operation creates data with the data timestamp as its value. The original data is not sent to the next components.

Timer

Alive Timer Operation

This operation sends the last received data to the next components only when no data for the same ID has been received before the timer expires.

Inform only once	select this to send non-updated data to the next components only once rather than at the end of each period.
Timer	period of the timer. This value cannot be lower than 5, as every 5 minutes a check is made to see if any ID isn't updated.

Conditions

A condition verifies a condition of the data before sending it to its actions or operations. Conditions can be stateful or stateless. In a stateless condition, new data is sent to the next component. A stateful condition sends the data only when the state is changed. For example, if state of ID A = true, data is sent only when the state is false.

Comparator

Constant Comparator Operation

Compares data with a constant value and adds the ADDED_THRESHOLD property containing the comparator in the output data. Data that meets the conditions is sent to the true actions/operations. Data that does not is sent to the false actions/operations.

Stateless	select this to send the result for each piece of data received, or do not select this to send data if a state changes for an ID.
Constant value	constant value used to compare data
Operator	operator of the comparison (<, <=, >, >=, = or !=)

Property Comparator Operation

Compares data with a numeric property contained in the data.

Stateless	select this to send the result for each piece of data received, or do not select this to send data if a state changes for an ID.
Property	must be a valid number to compare data with its value
Operator	operator of the comparison (<, <=, >, >=, = or !=)

Data Comparator Operation

Compares two pieces of data coming from two entries and defined by a grouping filter. Each comparison ends with the piece of data on the left sent to the true actions/operations when the condition is met, or to the false actions/operations if the condition is not met.

Stateless	select this to send the result for each piece of data received, or do not select this to send data if a state changes for an ID.
Operator	operator of the comparison (<, <=, >, >=, = or !=)
Left operand	must contain the name of the sender who will send the data that must be the left side of the operation.
Time limit	maximum delta between the two values in minutes
Group by	properties that create a logical group of values for this operation

String Comparator Operation

Compares one of the fields in the data with a custom string. Each piece of data with a condition met is sent to the true actions/operations, or to the false actions/operations if the condition is not met.

Stateless	select this to send the result for each piece of data received, or do not select this to send data if a state changes for an ID.
Field name	name of the field in the data to use in the comparison
Comparator	type of comparison (equals, contains, starts with or ends with)
Compare to	custom string used in the comparison

Counter

Data Counter Operation

This operation counts the number of data with the same ID and sends it to its true actions/operations when the count reaches the counter in the time limit.

Stateless	select this to send the result for each piece of data received, or do not select this to send data if a state changes for an ID.
Time range	time window boundary of the collected values in minutes
Count	counter

Grouped Data Counter Operation

This operation is useful to do an AND of previous conditions for data related by properties with different IDs. If the count of received related data matches the counter before the time limit, the last data is sent to its true actions/operations.

Stateless	select this to send the result for each piece of data received, or do not select this to send data if a state changes for an ID.
Time range	time window boundary of the collected values in minutes
Count	counter
Group by	a list of properties separated by white spaces that create a logical group of values for the operation

Sender Counter Operation

This operation counts the number of data from the sender and sends them to their true actions/operations when the count reaches the counter in the time limit.

Stateless	select this to send the result for each piece of data received, or do not select this to send data if a state changes for an ID.
Time range	time window boundary of the collected values in minutes
Count	counter

String Counter Operation

This conditional operation counts the times that data with the same ID has a property equal to a comparator, and sends them to their true actions/operations when the count reaches the counter in the time limit. To make a counter with more string operations, use a stateless String Comparator with a Sender Counter.

Stateless	select this to send the result for each piece of data received, or do not select this to send data if a state changes for an ID.
Time range	time window boundary of the collected values in minutes
Count	counter
Field name	name of the field in the data to use in the comparison
Compare to	custom string that must match the field value

Actions

Actions inform users of new or resolved problems in their system or run an auto resolving process.

Each alert data comes with a series of properties. Since most of the time they are coming from the Watch4net collectors, they are the same properties that you find when editing a report. When the Alerting Backend is running, you can have a view of those properties either in an entry point filter or when probing an active definition.

For example, suppose you want to send an email reporting the current CPU utilization for a specific CPU in a device based on the alert data below:

```
AlertData: {
  VALUE: 25
  TMST: 123456789 (unix timestamps)
  ID: uniqueId for this metric
  properties: {
    devtype: Array
    device: VPLEX1
    parttype: CPU
    part: CPU0
    name: CurrentUtilization
  }
}
```

You would add the following content:

Device PROP.'device' currently has a PROP.'name' on its PROP.'part' of VALUE.

This would translate dynamically as:

Device VPLEX1 currently has a CurrentUtilization on its CPU0 of 25.

Log

This action writes an entry in a log file. If two actions write to the same file, use the same parameters except for the content, because the Alerting Backend can't guarantee which configuration will be used. The default directory for the log files is Backends/Alerting-Backend/Default/logs.

File prefix	complete directory path and prefix of each log file. The prefix is followed by <code>_index.log</code> , where index is incremented until it reaches the retention parameter. Newest log files have the lowest index, so the current log is always <code>_0.log</code> .
Log retention	maximum number of log files kept.
Rotation time	optional if rotation-entry is used. Maximum time frame in minutes of the log content prior to its creation.
Rotation entries	optional if rotation-time is used. Maximum entries in a log file.
Entry content	content of the log entry. To add the value, date or ID of the current data, use the keywords VALUE, TMST or ID. To add a property of the current data, use the keyword PROP.'propertyName'.

Mail

This action sends an email when triggered.

To	comma-separated list of recipients
Subject	subject of the email. To add the value, date or ID of the current data, use the keywords VALUE, TMST or ID. To add a property of the current data, use the keyword PROP.'propertyName'.
Message	content of the email. To add the value, date or ID of the current data, use the keywords VALUE, TMST or ID. To add a property of the current data, use the keyword PROP.'propertyName'.

SNMP Trap

This action sends an SNMP trap when activated.

Host	receiver of the traps
Port	receiving port
Community	community of the traps
Generic ID	generic ID of the trap
Specific ID	ID of the trap format for the generic type
Trap content	comma-separated list of the content for each variable of the trap. A comma is escaped if it's preceded by a period. To add the value, date or ID of the current data, use the keywords VALUE, TMST or ID. To add a property of the current data, use the keyword PROP.'propertyName'. There's no limit to the number of variables in the trap content.

External Process

This action runs an external process on the Alerting Backend when triggered. To add the value, date or ID of the current data, use the keywords VALUE, TMST or ID. To add a property of the current data, use the keyword PROP.'propertyName'.

Command	command to run
Command parameters	optional. Comma-separated parameters of the command, for example -n 200. A comma is escaped if it's preceded by a period.
Environment parameters	optional. Comma-separated list of environment parameters of the terminal in key=value format (ex: JAVA_HOME=/opt/...). A comma is escaped if it's preceded by a period.

A

action	
log	42
mail	42
SNMP trap	43
adapter	19
alert	8
action	42
activate	26
adapter	19
arithmetic	37
condition	40
configuration	13, 14
counter	41
create	27
deactivate	26
definition	8, 27
display	9
engine	13
entry point filter	36
events adapter	20
example	23
file data adapter	20
grouped boxes	8, 29, 30
JVM	13
log	42
log template	24
logon	18
mail	42
operation	37
prerequisites	12
probe	31, 32
report data adapter	20
schedule report	15, 16, 17
SNMP trap	43
template	8, 21, 23
test	32
Tomcat server	14, 18
tools	18
trap clear template	25
trap notify template	25
values socket listener adapter	19
alert on scheduled report	8
Alerting Backend	8, 14, 31
Alerting Frontend	8, 14
arithmetic	
constant arithmetic	37
data arithmetic	37

C

condition	
comparator	40
counter	41

D

data creator	
properties to values creator	39
timestamp to value creator	39

E

entry point filter	36
events adapter	20

F

file data adapter	20
-------------------------	----

G

grouped boxes	8, 29
create	30

J

JVM	13
-----------	----

L

log template	24
--------------------	----

M

math	
configurable math	38
constant math	38

O

operation	
arithmetic	37
case	37
data creator	39
data repeater	38
date	38
math	38
no operation	37
previous polling operation	38
time	39
time window	38

P

probe	
alert	31, 32
prerequisites	31
report	34

R

report	
probe	34
report data adapter	20

S

schedule report	
alert	16, 17
SNMP trap	43

T

template	
alert	21
time	
alive timer operation	39
time window	
history operation	38
Tomcat server	14, 18
trap clear template	25
trap notify template	25

V

values socket listener adapter	19
--------------------------------------	----