OpenNMS Administrator Reference Guide

Configuration and Administration

1. Overview 1
1.1. About This Document 1
2. Configuration Files 2
2.1. actiond-configuration.xml 2
2.2. c3p0.properties 2
2.3. capsd-configuration.xml 2
2.3.1. the `capsd-configuration` tag 2
2.3.2. Protocol Plugins 3
2.4. categories.xml 4
2.4.1. Category Groups 4
2.4.2. Global Options 4
2.4.3. Categories 4
2.5. chart-configuration.xml 5
2.6. collectd-configuration.xml 6
2.7. create.sql, get*.sql, set*.sql 6
2.8. database-schema.xml 6
2.9. datacollection-config.xml 6
2.10. destinationPaths.xml 6
2.11. dhcpd-configuration.xml 6
2.12. discovery-configuration.xml 6
2.13. eventconf.xml 6
2.14. eventd-configuration.xml 6
2.15. events-archiver-configuration.xml 6
2.16. events.archiver.properties 7
2.17. groups.xml 7
2.18. http-datacollection-config.xml 7
2.19. javamail-configuration.properties 7
2.20. jcifs.properties 7
2.21. jmx-datacollection-config.xml 7
2.22. ksc-performance-reports.xml 7
2.23. linkd-configuration.xml 7
2.24. log4j.properties 7
2.25. log4j-controller.properties 7
2.26. magic-users.properties 7
2.27. map.disable 8
2.28. map.properties 8
2.29. model-importer.properties 8
2.30. monitoring-locations.xml 8
2.31. notifd-configuration.xml 8
2.32. notificationCommands.xml 8
2.33. notifications.xml 8
2.34. nsclient-config.xml 8
2.35. nsclient-datacollection-config.xml 8
2.36. opennms-datasources.xml 9
2.37. opennms-server.xml 9
2.38. opennms.properties 9
2.39. otrs.properties 9
2.40. poll-outages.xml 9
2.41. poller-config.properties 9

2.42. poller-configuration.xml 9
2.43. response-adhoc-graph.properties 9
2.44. response-graph.properties 9
2.45. rrd-configuration.properties 9
2.46. rtc-configuration.xml 9
2.47. scriptd-configuration.xml 10
2.48. service-configuration.xml 10
2.49. site-status-views.xml 10
2.50. snmp-adhoc-graph.properties 10
2.51. snmp-config.xml 10
2.52. snmp-graph.properties 10
2.53. statsd-configuration.xml 10
2.54. surveillance-views.xml 10
2.55. syslogd-configuration.xml 10
2.56. threshd-configuration.xml 10
2.57. thresholds.xml 10
2.58. tl1d-configuration.xml 10
2.59. translator-configuration.xml 11
2.60. trapd-configuration.xml 11
2.61. users.xml 11
2.62. vacuumd-configuration.xml 11
2.63. viewsdisplay.xml 11
2.64. vulnscand-configuration.xml 11
2.65. xmlrpcd-configuration.xml 11
2.66. xmpp-configuration.properties 11

Chapter 1. Overview

Admin Reference Overview

1.1. About This Document

This document covers the configuration of OpenNMS. In the following chapter we will cover each file and what it affects.

Chapter 2. Configuration Files

2.1. actiond-configuration.xml

Actions are external programs called based on events. This file controls the number of concurrent actions allowed as well as a time out for waiting on a return.

outstanding-actions	The maximum number of simultaneous processes launched by Actiond. If the number of launched processes currently running reaches this number, Actiond waits for a process to complete or get terminated before it launches the next process.
max-process-time	The maximum time that a launched process can take to complete. If execution time exceeds this time, the launched process is terminated.

Example 2.1. Example `actiond-configuration.xml`

Here is an example `actiond-configuration.xml`.

```
<?xml version="1.0"?>
<actiond-configuration
        max-outstanding-actions="10"
        max-process-time="120000">
</actiond-configuration>
```

2.2. c3p0.properties

C3P0 is the database connection pooling technology used by OpenNMS. In most cases you should never have to change this file. For details on configuration options, see the c3p0 configuration properties appendix in their documentation.

2.3. capsd-configuration.xml

This file defines capabilities (services) to be checked for discovered nodes.

2.3.1. the `capsd-configuration` tag

The <capsd-configuration> tag at the top of `capsd-configuration.xml` controls some basic behaviors of Capsd. The following attributes are accepted:

rescan-frequency	How long Capsd should wait before scanning existing nodes for changes in available services, in milliseconds. A value of `86400000` will cause Capsd to rescan nodes every 24 hours.
initial-sleep-time	How long Capsd should wait after OpenNMS starts before doing an initial scan of known nodes. A value of `30000` will cause Capsd to wait 30 seconds after startup before attempting to check known nodes for available services.
max-suspect-thread-pool-size	The maximum number of Java threads to use for scanning hosts upon receiving a `discovery/newSuspect` event.
max-rescan-thread-pool-size	The maximum number of Java threads to use for the regular rescan that occurs every `rescan-frequency` milliseconds.

abort-protocol-scans-if-no-route	Flag which determines Capsd's behavior in the event that a NoRouteToHost exception is generated during a protocol plugin scan of an interface. If true, Capsd will abort the protocol scanning process for the interface. Defaults to `false`.
delete-propagation-enabled	Determine if deleting an service propagates to deleting the interface if the service is the only one remaining on the interface. Likewise for the only interface on a node. Defaults to `true`.
xmlrpc	Flag which indicates if an external XMLRPC server has to be notified with any event process errors. Defaults to `false`.

2.3.2. Protocol Plugins

Protocol plugin entries define the services that are available for Capsd to detect. They can refer to any Java class that implements the OpenNMS `org.opennms.netmgt.capsd.Plugin` interface. A large number of default plugins are provided as a part of OpenNMS.

The following attributes are a part of the <protocol-plugin> tag:

protocol	The name of the protocol. This is a user-configurable value that should match similar service configuration entries in the `poller-configuration.xml` file. It is recommended that it only contain alphanumeric values. By convention, you can also append a dash and numbers, like so: "`HTTP-8080`" for common protocols listening on an alternate port.
class-name	The name of the Java class that implements the particular protocol.
scan	Whether or not to scan this particular protocol. If set to "off" Capsd will ignore the protocol plugin. Defaults to `on`.

2.3.2.2. Properties

A `protocol-plugin` entry can optionally contain zero or more `property` entries which allow specifying configuration to be passed as arguments to the plugin code.

2.3.2.3. Protocol Configuration

A `protocol-plugin` entry can optionally contain zero or more `protocol-configuration` tags which specify a subset of addresses for which the protocol should apply.

Each `protocol-configuration` entry can contain zero or more of the following ways of specifying addresses and ranges and of overriding other general `protocol-plugin` defaults:

specific	A single IP address.
range	A range of IP addresses.
property	A `protocol-plugin` key/value property which overrides the defaults that are part of the container `protocol-plugin` entry.
scan	Whether or not the addresses defined in the `specific` and `range` entries should be scanned. Defaults to `on`.

Example 2.2. Example `capsd-configuration.xml`

Here is an example `capsd-configuration.xml` that defines the `ICMP` protocol plugin, with a default timeout of 2000 milliseconds, and 1 retry. For the IP addresses 192.168.1.1 and 192.168.2.1 through 192.168.2.254, it will use a timeout of 1000 instead. Scanning is disabled for 192.168.1.2.

```
<?xml version="1.0"?>
<capsd-configuration
  rescan-frequency="86400000"
  initial-sleep-time="30000"
  max-suspect-thread-pool-size="6"
  max-rescan-thread-pool-size="3">

  <protocol-plugin protocol="ICMP" class-name="org.opennms.netmgt.capsd.plugins.IcmpPlugin" scan="on">
    <protocol-configuration>
      <specific>192.168.1.1</specific>
      <range>
        <begin>192.168.2.1</begin>
        <end>192.168.2.254</end>
      </range>
      <property key="timeout" value="1000" />
    </protocol-configuration>
    <protocol-configuration>
      <specific>192.168.1.2</specific>
      <scan>off</scan>
    </protocol-configuration>
    <property key="timeout" value="2000" />
    <property key="retry" value="1" />
  </protocol-plugin>
</capsd-configuration>
```

2.4. categories.xml

This file allows you to configure the categories of a device based on the mixture of services they run. You can also set expected error and warning levels for when availability drops below a certain percentage. These affect availability reports, as well as how the categories are displayed, if they are defined in `viewsdisplay.xml` as well.

2.4.1. Category Groups

Technically, the `categories.xml` file can have more than one "group" of categories. However, the <categorygroup> tag never got fleshed out in the code and it is recommended you do not use more than one <categorygroup> tag.

2.4.2. Global Options

Inside the category group, there are a number of options that can apply to all categories in the group.

name — One (and only one) <name> tag should be in each category group. This is the short name of the category group.

comment — One (and only one) <comment> tag should be in each category group. This is a longer description of the category group.

common — A common rule that will be applied to every category in the category group, in addition to that category's specific rule. The <common> tag should contain a single set of <rule> tags. For example:

```
<common>
 <rule><![CDATA[IPADDR != '0.0.0.0']]></rule>
</common>
```

2.4.3. Categories

Inside the category group, there can be a <categories> tag which can contain one or more <category> tags. The following tags are allowed in a category.

label	A descriptive label for the category. This needs to be unique across all categories, and will be displayed in the UI if the category is referenced in `viewsdisplay.xml`.
comment	An optional tag with a detailed comment describing the category.
normal	The normal threshold value for the category in percent. The UI displays the category in green if the overall availability for the category is equal to or greater than this value.
warning	The warning threshold value for the category in percent. The UI displays the category in yellow if the overall availability for the category is equal to or greater than this value but less than the normal threshold. If availability is less than this value, the category is displayed in red.
service	An optional tag specifying the service relevant to this category. For a node ID/IP address/service tuple to be added to a category, it will need to pass the category's rule and the service will need to be in the category service list. If there are no services defined, all tuples that pass the rule are added to the category.
rule	The rule that defines what nodes/interfaces/services match the category. It can match on services (Is*SERVICE*), IP addresses, and a number of other criteria.

Example 2.3. Example `categories.xml`

Here is an example `categories.xml` that defines a series of categories. It defines a global <rule> that makes sure that we get all IP devices in the 192.168 class B, and then individual categories which have more specific rules.

```
<catinfo>
 <header>
 </header>
 <categorygroup>
  <name>WebConsole</name>
  <comment>Service Level Availability by Functional Group</comment>
  <common>
   <rule><![CDATA[IPADDR IPLIKE '192.168.*.*']]></rule>
  </common>
  <categories>
   <category>
    <label><![CDATA[Overall Service Availability]]></label>
    <comment>This category reflects overall availability of every device on the 192.168.*.* network.</comment>
    <normal>99.99</normal>
    <warning>97</warning>
    <rule><![CDATA[IPADDR != '0.0.0.0']]></rule>
   </category>

   <category>
    <label><![CDATA[Pingable Routers]]></label>
    <comment>This category represents pingable routers with the "Router" service.</comment>
    <normal>99.99</normal>
    <warning>97</warning>
    <service>ICMP</service>
    <service>Router</service>
    <rule><![CDATA[(isICMP & isRouter)]]></rule>
   </category>
   <category>
    <label><![CDATA[VectaStar Elements]]></label>
    <comment>This category includes all managed interfaces which are related to the VectaStar system.</comment>
    <normal>99.99</normal>
    <warning>97</warning>
    <rule><![CDATA[(nodeSysOID like '.1.3.6.1.4.1.5419.1.3500.%')]]></rule>
   </category>
  </categories>
 </categorygroup>
</catinfo>
```

2.5. chart-configuration.xml

Used by some chart-generation code in the web UI.

2.6. collectd-configuration.xml

Configures various portions of data collection for RRD data.

2.7. create.sql, get*.sql, set*.sql

The `create.sql` file is the template for creating the OpenNMS database. It is used by the installer, and generally should not be modified. The get*.sql and set*.sql files define a series of stored procedures used in parts of the OpenNMS code.

2.8. database-schema.xml

This file is used internally by OpenNMS for the filtering system when formatting database queries. It is not meant to be modified by end-users.

2.9. datacollection-config.xml

This file is used for collecting data into round-robin databases for use in thresholding and graphing.

2.10. destinationPaths.xml

This file contains definitions for notifiation destination paths. It specifies whom to notify and by what method they should be notified.

2.11. dhcpd-configuration.xml

Configures the built-in DHCP daemon in OpenNMS (used for DHCP capability scanning and polling).

2.12. discovery-configuration.xml

This file defines the ranges of addresses to discover, (ping sweep) as well as time-outs, number of retries, and number of threads to dedicate to discovery. This file also provides initial-sleep-time and restart-sleep-time. These values are in milliseconds and control how long after OpenNMS is started, that discovery should begin it's initial pass through the addresses, and how long to wait between each additional pass.

2.13. eventconf.xml

This file defines the Universal Event Identifiers or UEIs as well as their event masks, descriptions, log messages, and severity levels.

2.14. eventd-configuration.xml

This file defines operating parameters for Eventd such as time outs and number of listener threads.

2.15. events-archiver-configuration.xml

Configuration for the event archiver daemon.

2.16. events.archiver.properties

Logging configuration for the event archiver daemon.

2.17. groups.xml

This file holds information about user groups, used for determining group membership for notifications.

2.18. http-datacollection-config.xml

This file defines configuration for collecting RRD data from HTTP and HTTPS.

2.19. javamail-configuration.properties

This file contains configuration for sending e-mail from OpenNMS for notifications, as well as the mail transport monitor.

2.20. jcifs.properties

This file defines configuration for accessing Windows SMB (CIFS) shares from OpenNMS.

2.21. jmx-datacollection-config.xml

This file contains configuration for datacollection using the Java Management eXtensions.

2.22. ksc-performance-reports.xml

This file contains configuration for KSC reports.

2.23. linkd-configuration.xml

Configuration for the link daemon, used to collect path/link information from routers for use in maps.

2.24. log4j.properties

This file defines logging information, including log size and rotation, as well as what level of logging should happen for different parts of the OpenNMS code.

2.25. log4j-controller.properties

???

2.26. magic-users.properties

This file includes special users, and takes precedence over users.xml for the users that it has information about. This is used internally to control permissions for certain functions that interact with the web UI.

2.27. map.disable

If this file exists, maps will not be enabled in the web UI.

2.28. map.properties

This file configures the SVG map feature.

2.29. model-importer.properties

Information used by the model importer.

2.30. monitoring-locations.xml

This file configures the monitoring locations used by the remote poller.

2.31. notifd-configuration.xml

Configures the notification daemon including auto-acknowledgement and notification queues.

2.32. notificationCommands.xml

This file defines how to accomplish various contact methods defined in `destinationPaths.xml`. This would include the location of executable and aliases for each type of contact along with any other information needed to send information of the specific type.

```
<command type="email">
 <name>/bin/mail</name>
 <lookup>email</lookup>
 <lookup>mail</lookup>
 <comment>for sending email notifications</comment>
 <argument streamed="false">
  <substitution>-s</substitution>
  <switch>-subject</switch>
 </argument>
 <argument streamed="false">
  <switch>-email</switch>
 </argument>
 <argument streamed="true">
  <switch>-tm</switch>
 </argument>
</command>
```

2.33. notifications.xml

This file defines which events or UEIs warrant notification and where the notifications should be sent and escalated to.

```
<notification name="nodeAdded">
 <uei><![CDATA[http://uei.opennms.org/nodes/nodeAdded]]></uei>
 <rule><![CDATA[IPADDR IPLIKE *.*.*.*]]></rule>
 <destinationPath>Email-Network/Systems</destinationPath>
 <text-message>OpenNMS has discovered a new node named %parm[nodelabel]%.    Please be advised.</text-message>
 <subject>%parm[nodelabel]% discovered.</subject>
</notification>
```

2.34. nsclient-config.xml

Configure the NSClient poller.

2.35. nsclient-datacollection-config.xml

Configure data collection through NSClient.

2.36. opennms-datasources.xml

This file defines information for accessing the OpenNMS database.

2.37. opennms-server.xml

Defines information about the management server.

2.38. opennms.properties

Sets global properties for the OpenNMS JVM.

2.39. otrs.properties

Configures properties for the OTRS ticketing plugin.

2.40. poll-outages.xml

Configures scheduled outages.

2.41. poller-config.properties

Configures various classes and settings used by the web UI relating to pollers.

2.42. poller-configuration.xml

This file is used to define packages as well as set up the various pollers. A package includes several items such as address ranges, services, outage calendars, and down time models.

2.43. response-adhoc-graph.properties

Settings related to graphing adhoc reports.

2.44. response-graph.properties

Settings related to making response-time graphs.

2.45. rrd-configuration.properties

Configuration for the RRD backend, including choosing RRDtool or JRobin for making round-robin files and graphing.

2.46. rtc-configuration.xml

This file defines properties for RTC (Real Time Console) such as the rolling window used to calculate percentages of down time, web UI refresh interval and how often RTC sends updates to the web interface.

2.47. scriptd-configuration.xml

Configure BSF languages used in Scriptd.

2.48. service-configuration.xml

This file defines opennms services to start. This is VM specific and controls which services are started in which VMs

2.49. site-status-views.xml

???

2.50. snmp-adhoc-graph.properties

Settings related to graphing adhoc SNMP reports.

2.51. snmp-config.xml

This file is used to define community strings for addresses or address ranges, one snmp-config entry per community.

2.52. snmp-graph.properties

This file is used to define RRD configurations for generating reports.

2.53. statsd-configuration.xml

Configuration for Statsd (Top N reports).

2.54. surveillance-views.xml

How many freaking *-view things do we have?!?

2.55. syslogd-configuration.xml

Configures the syslog daemon.

2.56. threshd-configuration.xml

Configure which thresholding packages used to trigger alerts when various thresholds are reached.

2.57. thresholds.xml

Configure thresholding groups based on RRD data.

2.58. tl1d-configuration.xml

Configure the TL1 daemon for interacting with telecom devices.

2.59. translator-configuration.xml

Translate incoming events based on a set of criteria.

2.60. trapd-configuration.xml

This file defines information for the OpenNMS SNMP trap daemon.

2.61. users.xml

This file holds information about users and their contact information. This information is used for authentication in the web UI, as well as notifications.

2.62. vacuumd-configuration.xml

Configure Vacuumd, a daemon which periodically runs a series of commands to keep OpenNMS running well (delete old events, clean up the database, etc.)

2.63. viewsdisplay.xml

This file defines layout of categories for display in the web UI. The actual categories are defined in the categories.xml file.

2.64. vulnscand-configuration.xml

Configure Vulnscand, the Nessus vulnerability scanning integration.

2.65. xmlrpcd-configuration.xml

Configure Xmlrpcd, used to be able to pass XML events back and forth between OpenNMS and external tools.

2.66. xmpp-configuration.properties

Configure the XMPP (Jabber) notification strategy.