

Copyright © 2005-2007 Tarus Balog, Matt Brozowski, David Hustace, Benjamin Reed

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections, with no Front-Cover Texts and with no Back-Cover Texts. A copy of the license is available at <http://www.gnu.org/copyleft/fdl.html>

Preface	iv
1. Introduction	1
1.1. Release 1.6.1	1
1.2. Release 1.6.0	1
2. What's New?	2
2.1. Changes in OpenNMS 1.6.1	2
2.1.1. New Features	2
2.1.2. Bug Fixes	2
2.1.3. Configuration/Behavior Changes	2
2.2. Changes in OpenNMS 1.6.0	4
2.3. Changes in OpenNMS 1.5.99	4
2.3.1. Bug Fixes	4
2.4. Changes in OpenNMS 1.5.98	4
2.4.1. Bug Fixes	4
2.5. Changes in OpenNMS 1.5.97	4
2.5.1. New Features	4
2.5.2. Bug Fixes	4
2.6. Changes in OpenNMS 1.5.96	5
2.6.1. Bug Fixes	5
2.7. Changes in OpenNMS 1.5.95	5
2.7.1. New Features	5
2.7.2. Bug Fixes	5
2.8. Changes in OpenNMS 1.5.94	5
2.8.1. New Features	5
2.8.2. Bug Fixes	6
2.8.3. Changes	6
2.9. Changes in OpenNMS 1.5.93	6
2.9.1. New Features	6
2.9.2. Bug Fixes	7
2.10. Changes in OpenNMS 1.5.92	7
2.10.1. New Features	7
2.10.2. Bug Fixes	7
2.11. Changes in OpenNMS 1.5.91	7
2.11.1. New Features	7
2.11.2. Bug Fixes	7
2.12. Changes in OpenNMS 1.5.90	8
2.12.1. Bug Fixes	8
2.13. Changes in OpenNMS 1.3.11	8
2.13.1. New Features	8
2.13.2. Bug Fixes	8
2.14. Changes in OpenNMS 1.3.10	8
2.14.1. New Features	8
2.14.2. Bug Fixes	9
2.15. Changes in OpenNMS 1.3.9	9
2.15.1. New Features	9
2.15.2. Bug Fixes	9
2.16. Changes in OpenNMS 1.3.8	9
2.16.1. New Features	9
2.16.2. Bug Fixes	10

2.17. Changes in OpenNMS 1.3.7	10
2.17.1. New Features	10
2.17.2. Bug Fixes	10
2.17.3. New Install Methods	10
2.18. Changes in OpenNMS 1.3.6	11
2.18.1. New Features	11
2.18.2. Bug Fixes	11
2.18.3. JNI Code Removal	11
2.19. Changes in OpenNMS 1.3.5	12
2.19.1. Bug Fixes	12
2.20. Changes in OpenNMS 1.3.4	12
2.20.1. New Features	12
2.20.2. Bug Fixes	12
2.21. Changes in OpenNMS 1.3.3	13
2.21.1. New Features	13
2.22. Changes in OpenNMS 1.3.2	13
2.22.1. Bug Fixes	13
2.22.2. New Features	13
2.23. Changes in OpenNMS 1.3.1	16
2.23.1. Bug Fixes	16
2.23.2. New Features	16
2.24. Changes in OpenNMS 1.3.0	16
2.24.1. Bug Fixes	16
2.24.2. New Features	16
3. Known Issues and Caveats	19
3.1. Current Release	19
3.2. Previous Releases	19
4. Supported Systems	20
4.1. Fully Supported	20
4.2. Unsupported	21

Preface

OpenNMS is the creation of numerous people and organizations, operating under the umbrella of the OpenNMS project. The original code base was developed and published under the GPL by the Oculan Corporation until 2002, when the project administration was passed on to Tarus Balog.

The current corporate sponsor of OpenNMS is [The OpenNMS Group](#), which also owns the OpenNMS trademark.

OpenNMS is a derivative work, containing both original code, included code and modified code that was published under the GNU General Public License. Please see the source for detailed copyright notices, but some notable copyright owners are listed below:

- Copyright © 2002-2008 [The OpenNMS Group, Inc.](#)
- Original code base for OpenNMS version 1.0.0 Copyright © 1999-2001 [Oculan Corporation](#).
- Mapping code Copyright © 2003 [Networked Knowledge Systems, Inc.](#)
- ScriptD code Copyright © 2003 [Tavve Software Company](#).

Please send any omissions or corrections to this document to [Tarus Balog](#).

Chapter 1. Introduction

About This Release

1.1. Release 1.6.1

Release 1.6.1 continues the 1.6 series with a small set of bug fixes and minor feature enhancements. It is a recommended upgrade for all users.

The codename for 1.6.1 is *bamboo army*.

1.2. Release 1.6.0

Release 1.6.0 is the first stable release in the OpenNMS 1.6 series.

It's been 3 and a half years since the last OpenNMS stable version, 1.2, was branched and released as production-ready. In that time, OpenNMS as a project has changed tremendously, the community has grown exponentially, and massive numbers of new features have been incorporated into the "unstable" 1.3.x series.

In that time, the unstable codebase solidified to the point that The OpenNMS Group supported it as if it were stable; it was *at least* as stable as 1.2.x was, but many users held off on upgrading because of the unstable moniker.

After a lot of work, and a renewed focus on getting the next stable release out the door, we are now prepared to declare OpenNMS 1.6 release-candidate-ready.

Why 1.6 instead of 1.4? 3 years is a lot of time, and a lot has happened in that time. We're not ready to call it 2.0, we want to redo the web UI first, but 1.4 didn't really do the massive changes since 1.2 justice. So: 1.6 it is.

Since it is a lot easier to do a release than it was in the 1.2 series (now that the native code is moved out into separate packages, and OpenNMS itself is distributed as pure-java sources), the goal is to continue to be on a much faster 6-month or year cycle for new releases.

Please, let us know if you have any problems at all in our [Bugzilla bug tracker](#).

Chapter 2. What's New?

Changes in This OpenNMS

2.1. Changes in OpenNMS 1.6.1

2.1.1. New Features

- KSC reports are now sorted by title. (Bug [#2817](#))
- You can now use \${ipaddr} in URLs for in the Page Sequence Monitor. (Bug [#2840](#))
- Support for new Fortinet traps was added. (Bug [#2847](#))
- A monitor for the trivial time service was added. (Bug [#2864](#))

2.1.2. Bug Fixes

- An error when attempting to delete URLs in the discovery admin page was fixed. (Bug [#2829](#))
- The poller was no longer setting the reason on nodeLostService events. (Bug [#2846](#))
- An exception in the path outages web UI when using PostgreSQL 8.3 was fixed. (Bug [#2866](#))
- A few issues with ticket and alarm triggers were fixed. (Bugs [#2868](#) and [#2869](#))
- A bug that would cause the OTRS ticketer plugin to ignore otrs.properties was fixed (Bug [#2870](#))
- An invalid regular expression in the thresholding configuration could cause thresholding not to happen and throw an exception. (Bug [#2873](#))
- The HTTP data collector now behaves properly when storeByGroup=true (Bug [#2875](#))
- The SSH monitor would cause a thread lock if a socket allowed connection but never returned any data. (Bug [#2896](#))
- The mail transport monitor would not retry during the read test. (Bug [#2897](#))

2.1.3. Configuration/Behavior Changes

A number of changes were done to configs to clean up usage and get rid of old unnecessary tags.

- `capasd-configuration.xml`: `capasd-configuration.xml` no longer requires the `user-defined` attribute. (Bug [#2872](#))
- `eventconf.xml`: the `<logmsg>` tag now accepts an optional attribute, `notify`, which can be set to `false` to unilaterally suppress any notifications on that event. (Bug [#2891](#))
- `http-datacollection-config.xml`: The `name` attribute on URI tags in the HTTP data collection configuration file is now required. It was common practice to do so anyways, but you will need to check your configs to make sure it is defined. (Bug [#2876](#))

- `http-datacollection-config.xml`: The `url` tag can now take a number of options relating to how regular expressions should be parsed. (Bug [#2882](#))

[canonical-equivalence](#)

When this flag is specified then two characters will be considered to match if, and only if, their full canonical decompositions match. The expression `"a\u030A"`, for example, will match the string `"å"` when this flag is specified. By default, matching does not take canonical equivalence into account.

[case-insensitive](#)

Enables case-insensitive matching. By default, case-insensitive matching assumes that only characters in the US-ASCII charset are being matched. Unicode-aware case-insensitive matching can be enabled by specifying the `unicode-case` flag in conjunction with this flag.

[comments](#)

Permits whitespace and comments in pattern. In this mode, whitespace is ignored, and embedded comments starting with `#` are ignored until the end of a line.

[dotall](#)

Enables `dotall` mode. In `dotall` mode, the expression `.` matches any character, including a line terminator. By default this expression does not match line terminators.

[literal](#)

Enables literal parsing of the pattern. When this flag is specified then the input string that specifies the pattern is treated as a sequence of literal characters. Metacharacters or escape sequences in the input sequence will be given no special meaning. The flags `case-insensitive` and `unicode-case` retain their impact on matching when used in conjunction with this flag. The other flags become superfluous.

[multiline](#)

Enables multiline mode. In multiline mode the expressions `^` and `$` match just after or just before, respectively, a line terminator or the end of the input sequence. By default these expressions only match at the beginning and the end of the entire input sequence.

[unicode-case](#)

Enables Unicode-aware case folding. When this flag is specified then case-insensitive matching, when enabled by the `case-insensitive` flag, is done in a manner consistent with the Unicode Standard. By default, case-insensitive matching assumes that only characters in the US-ASCII charset are being matched.

[unix-lines](#)

Enables Unix lines mode. In this mode, only the `\n` line terminator is recognized in the behavior of `.`, `^`, and `$`.

- `notifd-configuration.xml`: The default behavior of the numeric page (`-nm`) field in notifications has been changed to prepend `"RESOLVED:"` to the message, just like other page types, since most people are using the numeric field for SMS pages. To restore the old behavior, add the `skip-numeric-resolution-prefix` to the global `<notifd-configuration>` tag at the top of the file, set to `"true"`.

The full list of changes is available in [bugzilla](#).

2.2. Changes in OpenNMS 1.6.0

OpenNMS 1.5.99 passed release candidate testing and was re-tagged as 1.6.0.

2.3. Changes in OpenNMS 1.5.99

2.3.1. Bug Fixes

- A bug in the poller which caused the reason code not to be associated with node lost service events was fixed. (Bug [#2846](#))

2.4. Changes in OpenNMS 1.5.98

2.4.1. Bug Fixes

- A bug in the OTRS integration was fixed. (Bug [#2835](#))
- A critical bug in the poller and HTTP monitor which could cause services from being restored in some cases was fixed. (Bugs [#2841](#) and [#2842](#))

2.5. Changes in OpenNMS 1.5.97

2.5.1. New Features

- When searching for a node, if the search returns only a single node, it will redirect directly to the node page. (Bug [#1518](#))
- Support for new Ricoh/Savin Printer data collections was added. (Bug [#1941](#))
- The default thresholding behavior has been changed to use the new in-line collectd-based thresholder. (Bug [#2611](#))
- The comment field in assets no longer has a size limit. (Bug [#2763](#))
- The default logging level is now WARN instead of DEBUG. (Bug [#2796](#))
- PostgreSQL 8.3 is now officially supported in the installer. (Bug [#2808](#))

2.5.2. Bug Fixes

- A bug in the scheduled outages UI was fixed. (Bug [#2670](#))
- A number of thread concurrency issues have been fixed. (Bug [#2715](#))
- The HTTP collector now handles string attributes correctly. (Bug [#2806](#))
- A few other bugs have been fixed since 1.5.96, in the process of preparing for 1.6.0. A full list of bugs fixed in the release can be found in [bugzilla](#).

2.6. Changes in OpenNMS 1.5.96

2.6.1. Bug Fixes

- An exception introduced into the notification browser as part of the XSS security fixes was made unexceptional. (Bug [#2776](#))

2.7. Changes in OpenNMS 1.5.95

2.7.1. New Features

- Nodes in category list are now sorted by name. (Bug [#2708](#))

2.7.2. Bug Fixes

A bug in the poller that would cause failed services to be shown as available was fixed. (Bug [#2769](#))

A few other bugs have been fixed since 1.5.94, in the process of preparing for 1.6.0. A full list of bugs fixed in the release can be found in [bugzilla](#).

A number of security issues have been fixed. (Bug [#2760](#))

2.8. Changes in OpenNMS 1.5.94

2.8.1. New Features

- OpenNMS now has provisional support for PostgreSQL 8.3. All known bugs have been fixed, but there may be some corner cases left that aren't covered by unit tests. To use OpenNMS with PostgreSQL 8.3, you will need to run the install tool with the "-Q" option. (Bug [#2613](#))
- OpenNMS now automatically writes a thread dump to output.log on shutdown, for easier debugging. (Bug [#2181](#))
- SNMP parameters can now be set in the Collectd configuration. (Bug [#2195](#))
- Relative thresholds can now use negative numbers. Additionally, you can now threshold on absolute changes as well (loss in dB on fiber links, etc.) (Bugs [#2275](#) and [#2604](#))
- XmlRpcNotifier now allows a timeout parameter. (Bug [#2342](#))
- The java web start remote poller can now be run in a headless mode. See [this page](#) for configuration details. (Bug [#2354](#))
- Asset comments now retain formatting. In addition, the exporter creates real CSV files, so you can have asset data with commas in them. ;) (Bug [#2363](#))
- The OSS/J QoS interface has been enhanced in a number of ways, mostly related to documentation and simplifying importing. See [bug #2618](#) for details.
- A new ticketing plugin for [OTRS](#) has been added. (Bug [#2658](#))
- Initial support was added for interfacing with devices that speak Transactional Language 1, or TL-1. (Bug [#2693](#))
- The HTTP Collector can now substitute IP addresses in URLs it's collecting from (Bug [#2590](#))

- Support for new Aedilis, Cisco, Citrix, Compaq, McAfee, Polycom, Radlan, and UPS-MIB traps (Bugs [#2511](#), [#2542](#), [#2554](#), [#2566](#), [#2581](#), [#2598](#), [#2599](#), [#2643](#), [#2671](#), and [#2722](#))
- Support for new Brocade, Citrix, Dell, Fortinet, HP-UX, Kyocera, Liebert, Novell, and SNMP Informant data collections (Bugs [#2370](#), [#2371](#), [#2391](#), [#2511](#), [#2579](#), [#2624](#), [#2629](#), [#2663](#), [#2686](#), and [#2700](#))

2.8.2. Bug Fixes

A rare but insidious bug in the poller that caused some outages to not be properly resolved was fixed. (Bug [#2702](#))

A large number of other bugs have been fixed since 1.5.93, in the process of preparing for 1.6.0. A full list of bugs fixed in the release can be found in [bugzilla](#).

A number of cross-site scripting (XSS) security issues have been fixed. (Bugs [#2631](#), [#2633](#), and [#2634](#))

2.8.3. Changes

In the plugins for capsd and the monitors for the service poller, the correct parameter to use for retries is "retry" however some used "retries". This has been corrected, but on upgrading one may want to change the configuration files if they have been customized. Those changed were:

- service poller monitors:
 - SNMP Monitor
 - OMSA Storage Monitor
 - PERC Monitor
 - Disk Usage Monitor
- capsd plugins:
 - NTP Plugin
 - DNS Plugin

The default RRD roll-up schedule for data collected using the NSClient Collector has been changed to match that used for some time by the SNMP Collector. When upgrading to release 1.5.94, users who rely on NSClient data collection and do not wish to lose any historical data must take care to preserve the existing roll-up schedule in the nsclient-datacollection-config.xml file. Users who are not concerned with keeping historical NSClient performance data can adopt the new roll-up schedule, but will need to delete all RRD files created while the old roll-up schedule was in effect. Failure to do so will result in new data not being stored.

2.9. Changes in OpenNMS 1.5.93

2.9.1. New Features

This is a quick bugfix release, no new feature were added.

2.9.2. Bug Fixes

This release is intended primarily to solve an issue in SNMP4J which could cause performance issues from excessive logging in some situations. (Bug [#2552](#))

In addition, the installer was updated to not force a specific amount of memory when run, so it should work with large databases without the need to edit the upgrade script.

2.10. Changes in OpenNMS 1.5.92

2.10.1. New Features

Xmlrpcd has been improved to allow multicasting events as well as a few other changes. (Bug [#1487](#))

Support for logging logins, failed login attempts, logouts, and session timeouts was added. (Bug [#1580](#))

Support for new Juniper, Cisco, Sonus, and VMware traps (Bugs [#2368](#), [#2487](#), [#2500](#), [#2502](#), and [#2540](#))

Support for new Asterisk, Juniper, Cisco, Windows, Mikrotik, and UCD data collections (Bugs [#2356](#), [#2367](#), [#2384](#), [#2387](#), [#2390](#), [#2420](#), [#2477](#), [#2483](#), and [#2529](#))

There is a new parameter that can be used to extract asset information for use in events and notices. The parameter `%asset[fieldname]%` will look up the field in the assets table represented by "fieldname" for the nodeid in the event. For example: `%asset[description]%` will return the description field. If a field is empty or not available it will return "Unknown". (Bug [#2465](#))

2.10.2. Bug Fixes

A full list of bugs fixed in the release can be found in [bugzilla](#).

2.11. Changes in OpenNMS 1.5.91

2.11.1. New Features

The scheduled outage web UI improvements that were partially implemented in 1.5.91 are finished. You can now create daily scheduled outages.

OpenNMS XML configuration files will now be validated if xmllint is available at startup.

2.11.2. Bug Fixes

A number of exceptions and other code errors have been fixed.

A few issues relating to thresholding and data collection have been resolved.

A full list of bugs fixed in the release can be found in [bugzilla](#).

2.12. Changes in OpenNMS 1.5.90

2.12.1. Bug Fixes

NSClient and HTTP collections used to need a "nsclient-collection" and "http-collection" parameter (respectively) in collectd-configuration.xml. This is now supported, but deprecated, and the name changed to "collection", as is the case for SNMP collection.

Data collection from systems supporting the UCD-SNMP SYSSTAT MIB group now includes the raw counters for context and interrupts. The deprecated one-minute gauge versions of these metrics are still collected if the agent provides them, but their resource graphs are no longer in the default list.

A full list of bugs fixed in the release can be found in [bugzilla](#). There were over 70 of them.

In addition, a number of new event definitions were added. Note that the UEIs for Foundry Networks now all read "foundry" whereas before some read "foundry" and others read "FoundryNetworks". This will affect any notifications based on those UEIs.

2.13. Changes in OpenNMS 1.3.11

2.13.1. New Features

There are no new features in 1.3.11, but there was one that was left out of the description for 1.3.10 that should be mentioned.

The [Mail Transport Monitor](#) tests a mail server by actually sending an e-mail and checking that it was received. This can be a very useful for measuring the response time of mail systems.

2.13.2. Bug Fixes

A lot of changes were made under the covers to the eventd process in 1.3.10. One of those changes broke events from interfaces with an IP address that exists on more than one node. While not a problem that the average person is likely to encounter, it was enough of an issue to warrant a fix.

2.14. Changes in OpenNMS 1.3.10

2.14.1. New Features

OpenNMS now supports an integration with the [Hyperic HQ](#) agent. There is a detailed [white paper](#) available that describes the necessary configuration changes.

The classes used to discover and poll NRPE (Nagios Remote Plugin Execution) checks from OpenNMS now support the [NRPE protocol over secure sockets layer \(SSL\)](#). Since both the official NRPE addon from Nagios and the NRPE listener of NSClient++ use SSL by default, the new default in OpenNMS is to use SSL for NRPE.

OpenNMS now supports [discovery and monitoring of Windows services](#) (even those that do not provide network services, such as the Task Scheduler) via the SNMP service that ships with Windows.

[Thresholding](#) has been (optionally) moved into the collection daemon. Changes to your configuration is required to use this new code, but once done, thresholds are calculated on data as it is collected.

This is significantly more efficient (data is not read back from the RRD/JRB files on disk), and also removes the need for the "range" parameter.

It is now possible to relocate the /var/opennms and /var/log/opennms when installing the RPMs, using the "--relocate" option to RPM.

2.14.2. Bug Fixes

A large number of bugs were addressed in this release. Bugzilla contains [the complete list](#). Also, a large number of changes were made under the covers to such processes as eventd.

2.15. Changes in OpenNMS 1.3.9

2.15.1. New Features

There are a couple of new features in this release:

- There is now a way to add and delete groups through the webUI.
- Users can change their password by clicking on their username in the web page header.
- Added event definitions for Oracle and ISS events.

2.15.2. Bug Fixes

In certain cases the number of allowed database connections could be completely exhausted, causing discovery and notifications to stop working. This has been corrected.

2.16. Changes in OpenNMS 1.3.8

2.16.1. New Features

There are a number of new features in this release:

- RSS Feeds now exist for outages, alarms and notifications.
- Outage duration is shown in the outage list.
- KSC Reports now allow for the number of graphs per line to be specified per report.
- A new SSH monitor has been created to improve on the old one based on the TCP monitor.
- Added a "strict timeout" option to monitors to enable connection errors to be retried after the timeout expires versus immediately.

In order to use the "strict timeout" option, simply add it as a parameter to your monitor config in poller-configuration.xml:

```
<service name="SMTP" interval="300000" user-defined="false" status="on">
  <parameter key="retry" value="1"/>
  <parameter key="timeout" value="3000"/>
  <parameter key="strict-timeout" value="true"/>
  <parameter key="port" value="25"/>
  <parameter key="rrd-repository" value="/opt/opennms/share/rrd/response"/>
```

```
<parameter key="rrd-base-name" value="smtp"/>
<parameter key="ds-name" value="smtp"/>
</service>
```

Probably the biggest new feature is the ability to run on Windows. We are testing an installer now and will be releasing it soon.

2.16.2. Bug Fixes

As OpenNMS moves closer to the next stable release, the number of features will decrease as bug fixes increase. This release is no different, with [lots of bugs](#) corrected in this version.

2.17. Changes in OpenNMS 1.3.7

2.17.1. New Features

Jetty used instead of Tomcat: OpenNMS now runs with its own servlet container using jetty. The default OpenNMS port has been moved from 8080 to 8980. Tomcat is not required but can still be used if desired.

StrafePing: Based on Tobi Oeticker's [SmokePing](#), this new monitor will report on ping latency in a nifty new graph. It will send out 20 pings every five minutes by default. The monitor can be configured to alert on any number of missed pings as well.

New Zoom: The zoom code borrowed from Cacti had issues with offsets. The zoom code from SmokePing was brought in to replace it. Now all the offsets work well except for Internet Explorer. This should be corrected for IE in the next release.

Jira Plugin: Thanks to Johan, there is now a Jira trouble ticket plugin to go with the one for [CentricCRM](#).

2.17.2. Bug Fixes

Performance: Data collection performance has been greatly improved. On one site OpenNMS was able to collect on 15 variables from 126,000 interfaces on 4467 devices in less than a minute with 150 collection threads. Storing this information was a different matter, since it really depends on the performance of the disk subsystem. OpenNMS will queue the data to be written, so while no data is lost it may take some time before a particular file to be updated. In the past this resulted in OpenNMS being "behind" - i.e. the last update shown on a graph could be several polling cycles old. Since, in the above example, only a few of those 126K interfaces would be viewed at any one time, new promotion code was added to the webUI that will update a file with pending data before it is displayed, thus removing that shortcoming. Also, with the removal of Tomcat as a requirement, the whole application runs in one VM. This, along with some code changes, will show a marked speed improvement in the webUI.

The complete list of bug fixes in this release can be found in [bugzilla](#).

2.17.3. New Install Methods

With the removal of the JNI code in release 1.3.6, this has enabled the creation of yum and apt repositories for a number of operating systems (including a nightly snapshot for the brave). There is a [Quickstart](#) page on the wiki describing these new methods for managing OpenNMS packages and upgrades.

2.18. Changes in OpenNMS 1.3.6

OpenNMS 1.3.6 is the next step in the accelerating release schedule toward OpenNMS 2.0. This release is being tagged just before the annual Dev-Jam developer's conference to provide a good launch point for the development that will happen this week.

2.18.1. New Features

There is a new Data Export feature that will export performance data stored in the "RRD" files to an XML file that can be used by other application (currently only available if you are using JRobin to store the data). This feature is accessed by going to a particular URL (/opennms/summary/results.htm) and passing four parameters: filterRule: a filter rule This is any filter that normally works in OpenNMS, such as "ipaddr iplike *.*.*.*" (although I'd worry about using that one with a open attributeSieve). startTime (in seconds since epoch) i.e. this is in "timeticks" or the value returned with "date +%s" endTime (in seconds since epoch) ditto "timeticks" attributeSieve=a regexp that can be passed to String.match() to match the attribute name Thus ".*" would return all values from all RRDs for the nodes that match the filter above. Here's an example using wget:

```
wget --http-user=admin --http-passwd=admin -O summary.xml \
'http://172.16.8.100:8180/opennms/summary/results.htm?filterRule=ipaddr+iplike+10.136.123.1&startTime=1184173183&endTime=1185219010&attributeSieve=.*'
```

This will return the summary.xml file for all of the variables being collected for the node that contains 10.136.123.1. Here's one for ifInOctets and ifOutOctets for the whole system:

```
wget --http-user=tarus.balog --http-passwd=secret -O summary.xml \
'http://172.16.8.100:8180/opennms/summary/results.htm?filterRule=ipaddr+iplike+*.*.*.*&startTime=1184173183&endTime=1185219010&attributeSieve=(ifInOctets|ifOutOctets)'
```

Speaking of JRobin, some changes have been made to make the graphs prettier (thanks to Will Fraley).

2.18.2. Bug Fixes

The data collection code has been tweaked yet again to improve performance.

2.18.3. JNI Code Removal

This release is the first OpenNMS without any JNI code in the OpenNMS application or webapp. Previously, OpenNMS needed to be compiled on each platform due to slight differences in libraries for the C code used to implement iplike (the PostgreSQL function for parsing IP addresses), jrrd (the code to integrate OpenNMS with RRDTool) and jicmp (the code that let's OpenNMS send pings). Thus these functions must be installed separately. While OpenNMS will work without jrrd (it will use JRobin to store data), the iplike function should be installed for performance reasons (a pg_sql version will be used if iplike is not available), and jicmp is pretty much required.

There are a number of ways to get these functions installed. On Sourceforge there are RPMs for many distros and as well as a source tarball. The source tarball can be installed using the usual "configure; make; make install" method used for most C code. If the operating system supports RPMs, then "rpmbuild -tb [tarball]" will create a binary RPM if it is not already available.

In order to make OpenNMS aware of where the code is located, a new option has been added to the installer: -l

Since the default location for these files is "/usr/local/lib" this installer would be run similar to:

```
/opt/opennms/bin/install -l /usr/local/lib -dis
```

2.19. Changes in OpenNMS 1.3.5

2.19.1. Bug Fixes

The reason for this release is a bug involving notifications. Due to the move toward database independence some code was introduced that caused notifications on events with a serviceid of null (such as those from traps) to fail. The original problem was misdiagnosed and the attempt to fix it, version 1.3.4, made the problem worse. It has been correctly resolved in this release. In addition, some major changes were made to the way OpenNMS data collection is started in order to use a smaller memory footprint and provide better performance.

2.20. Changes in OpenNMS 1.3.4

2.20.1. New Features

The only new feature is the ability to invert the result of a monitor. By adding the parameter "invert-status" to a monitor, up becomes down, and down becomes up. Useful for monitoring things, like an ISDN link, that indicate a problem when they are available. This was available for ICMP in 1.2.x, but it has now been ported to deal with all monitors.

An example of how to use this is as follows. Suppose there are a number of IP addresses on the network that represent ISDN backup lines. Thus a successful "ping" to those IPs means that there is a problem with the network. In other words, up is down.

First, there needs to be a service in `capsd-configuraion.xml` that defines the "not" ICMP service:

```
<protocol-plugin protocol="NotICMP" class-name="org.opennms.netmgt.capsd.plugins.IcmpPlugin" scan="off" user-defined="false">
  <property key="timeout" value="2000" />
  <property key="retry" value="1" />
</protocol-plugin>
```

Note that the scan for this service is "off". There are a number of ways to add services to IP addresses, but an easy one is to use the `send-event.pl` script to send an event to add the service directly:

```
send-event.pl uei.opennms.org/internal/capsd/updateService localhost \
--interface 172.20.1.12 \
--service NotICMP \
--parm 'nodelabel kenny.opennms.com' \
--parm 'action ADD'
```

This will add the NotICMP service to the node "kenny.opennms.com" on interface "172.20.1.12". Repeat for all other ISDN links.

Finally the poller has to be told to poll this service. Edit `poller-configuration.xml` and add:

```
<service name="NotICMP" interval="300000" user-defined="false" status="on">
  <parameter key="invert-status" value="true"/>
  <parameter key="retry" value="2"/>
  <parameter key="timeout" value="3000"/>
  <parameter key="rrd-repository" value="/opt/opennms/share/rrd/response"/>
  <parameter key="ds-name" value="noticmp"/>
</service>
```

and remember to put the monitor at the bottom:

```
<monitor service="NotICMP" class-name="org.opennms.netmgt.poller.monitors.IcmpMonitor"/>
```

The key here is the "invert-status" property. This should work with every monitor in the application.

2.20.2. Bug Fixes

The reason for this release is a bug involving notifications. In 1.3.3 the ability to match on severities was introduced. For example, you could send a notice on any even that was "Critical". Unfortunately,

there was an unfortunately long "if" statement that got pooched when this feature was added. It blocked notifications based on traps. In addition, there was a bug with packages with just <specific> tags (no include-range tags) passing all IP addresses that has been addressed.

2.21. Changes in OpenNMS 1.3.3

2.21.1. New Features

In an attempt to get the release out as fast as possible, we've decided to track new features on the OpenNMS wiki's [New and Noteworthy](#) page.

However, there are two main "gotchas" of which to be aware.

The first is that the default RRD file storage has changed. If you want to use the new format, you will need to delete your old RRD/JRobin data. OpenNMS was storage an incredibly large amount of data, especially compared to projects like MRTG. It will now store "as polled" 5 minute samples for a week, hourly samples for two months and daily MIN, MAX and AVERAGE for a year. This reduces the size of the RRD files by more than 80%.

The second one is that the installer no longer installs the Postgresql "iplike" function. It needs to be installed as a separate package (if it already exists in the running installation of Postgresql, then nothing else needs to be done). If the installer does not find iplike it will install a platform independent psql version. That version will work but at a lower performance than the native C code version.

2.22. Changes in OpenNMS 1.3.2

2.22.1. Bug Fixes

SNMP interface descriptions are no longer printed in hex if they contain unprintable characters. In particular, this was seen on Microsoft Windows platforms with the default Windows SNMP daemon.

2.22.2. New Features

2.22.2.1. NSClient support

An [NSClient](#) poller and capsd plugin have been contributed by [Matt Raykowski](#). This allows performance counters on Microsoft Windows platform to be queried, with the help of the NSClient agent software on the Windows side. No data collection is performed at this time.

For testing, you can run the NsclientManager implementation directly:

```
java -cp $OPENNMS_HOME/lib/opennms_services.jar <hostname> <command> <criticalPercent> <warningPercent> <parameter>
```

Here are a couple of example tests using the command-line tool:

1. Testing the client version:

```
java -cp $OPENNMS_HOME/lib/opennms_services.jar org.opennms.netmgt.poller.nsclient.CheckNsc <hostname> CLIENTVERSION 0 0 "1.0.7.0"
```

2. Testing NT services:

```
java -cp $OPENNMS_HOME/lib/opennms_services.jar org.opennms.netmgt.poller.nsclient.CheckNsc <hostname> USEDDISKSPACE 95 90 "C:"
```

3. Testing memory usage:

```
java -cp $OPENNMS_HOME/lib/opennms_services.jar org.opennms.netmgt.poller.nsclient.CheckNsc <hostname> MEMUSE 95 90 ""
```

Available commands:

CLIENTVERSION

This check uses the parameter property only. This string must be formatted like "#.#.#" or empty. If you provide a number, such as "1.0.7.0", you are specifying the minimum version supported, any version lower than this parameter will return 'critical'. If you do not specify a parameter (for example, supplying "") the check always returns 'okay' and will provide the version number in the response.

Using CLIENTVERSION in capsd:

```
<protocol-plugin protocol="NSC-CLIENTVERSION" class-name="org.opennms.netmgt.capsd.plugins.NscClientPlugin" scan="on" user-defin
  <property key="port" value="1248" />
  <property key="timeout" value="3000" />
  <property key="retry" value="2" />
  <property key="command" value="CLIENTVERSION" />
  <!-- parameter is optional, if you want to do version checking
  -->
  <property key="parameter" value="1.0.7.0" />
</protocol-plugin>
```

CPULOAD

This check uses the warningPercent and criticalPercent properties only. These values must be integers in percentage format.

Using CPULOAD in capsd:

```
<protocol-plugin protocol="NSC-CPULOAD" class-name="org.opennms.netmgt.capsd.plugins.NscClientPlugin" scan="on" user-defin
  <property key="port" value="1248" />
  <property key="timeout" value="3000" />
  <property key="retry" value="2" />
  <property key="command" value="CPULOAD" />
  <property key="warningPercent" value="90" />
  <property key="criticalPercent" value="95" />
</protocol-plugin>
```

UPTIME

This check simply returns the uptime of the system. No validation is performed.

USEDISKSPACE

This check uses the warningPercent and criticalPercent properties to determine thresholds. These values must be integers and they should be in percentage form (e.g. 1-100.) The parameter property is used to determine the drive letter, either "C" or "C:" can be used. Some older services have mixed support, so try adding/removing the ':' if you are experiencing problems.

Using USEDISKSPACE in capsd:

```
<protocol-plugin protocol="NSC-C-SPACE" class-name="org.opennms.netmgt.capsd.plugins.NscClientPlugin" scan="on" user-defin
  <property key="port" value="1248" />
  <property key="timeout" value="3000" />
  <property key="retry" value="2" />
  <property key="command" value="USEDISKSPACE" />
  <property key="warningPercent" value="90" />
  <property key="criticalPercent" value="95" />
  <property key="parameter" value="C:" />
</protocol-plugin>
```

SERVICESTATE

This check determines the status of NT services on a remote server. This check only uses the parameter property. The parameter property should contain a comma delimited list of services you would like the status of.

Using SERVICESTATE in capsd:

```
<protocol-plugin protocol="NSC-SERVICE-SERVERS" class-name="org.opennms.netmgt.capsd.plugins.NscClientPlugin" scan="on" user-defin
```

```
<property key="port" value="1248" />
<property key="timeout" value="3000" />
<property key="retry" value="2" />
<property key="command" value="SERVICESTATE" />
<property key="parameter" value="Eventlog,lammanserver,Netlogon,RpcSs" />
</protocol-plugin>
```

PROCSTATE

This check determines whether or not a list of processes are running on a remote server. This check only uses the parameter property. The parameter property should contain a comma delimited list of processes you want to determine the status of.

Using PROCSTATE in capsd:

```
<protocol-plugin protocol="NSC-PROCESS-NAVISPHERE" class-name="org.opennms.netmgt.capsd.plugins.NsclientPlugin" scan="on" user-defined="true">
  <property key="port" value="1248" />
  <property key="timeout" value="3000" />
  <property key="retry" value="2" />
  <property key="command" value="SERVICESTATE" />
  <property key="parameter" value="naviagent.exe,EmcPowSrv.exe" />
</protocol-plugin>
```

MEMUSE

This check uses the warningPercent and criticalPercent properties only. These values must be integers in percentage format.

Using MEMUSE in capsd:

```
<protocol-plugin protocol="NSC-MEMORY" class-name="org.opennms.netmgt.capsd.plugins.NsclientPlugin" scan="on" user-defined="true">
  <property key="port" value="1248" />
  <property key="timeout" value="3000" />
  <property key="retry" value="2" />
  <property key="command" value="MEMUSE" />
  <property key="warningPercent" value="90" />
  <property key="criticalPercent" value="95" />
</protocol-plugin>
```

COUNTER

This check is used to check the PerfMon OID objects on a remote Windows server. This check uses the warningPercent and criticalPercent properties as *values*. These values may or may not actually be in percentage format. You will have to use discretion when setting these up and read the documentation for the specific PerfMon OID that you will be monitoring. It also uses the parameter property to define the PerfMon OID that you will be monitoring.

Using COUNTER in capsd:

```
<protocol-plugin protocol="NSC-PRINT-QUEUE" class-name="org.opennms.netmgt.capsd.plugins.NsclientPlugin" scan="on" user-defined="true">
  <property key="port" value="1248" />
  <property key="timeout" value="3000" />
  <property key="retry" value="2" />
  <property key="command" value="COUNTER" />
  <property key="warningPercent" value="10" />
  <property key="criticalPercent" value="20" />
  <property key="parameter" value="\Print Queue(_Total)\Jobs" />
</protocol-plugin>
```

FILEAGE

This check is used to determine the age of a specified file on the remote server. This check uses the warningPercent and criticalPercent properties as *values*. These values will be the newest age of the file allowed in *minutes*. For example, if you set the criticalPercent to 60 then a file that is 59 minutes or *newer* will result in a critical response from the manager. This check uses the parameter property to determine the *full path* of the file to be monitored. This check is useful for monitoring log files or other files that may collect critical events or files that should rarely change. Note: This check does not yet support using the modified date response supplied by the server.

Using FILEAGE in capsd:

```
<protocol-plugin protocol="NSC-SYSTEMFILES" class-name="org.opennms.netmgt.capsd.plugins.NsclientPlugin" scan="on" user-
  <property key="port" value="1248" />
  <property key="timeout" value="3000" />
  <property key="retry" value="2" />
  <property key="command" value="FILEAGE" />
  <property key="warningPercent" value="0" />
  <property key="criticalPercent" value="525600" /> <!-- two years -->
  <property key="parameter" value="C:\autoexec.bat" />
</protocol-plugin>
```

2.23. Changes in OpenNMS 1.3.1

2.23.1. Bug Fixes

A potential database connection leak was discovered in 1.3.0 and this release was generated to include that fix.

2.23.2. New Features

There are no new features in 1.3.1, but more work was done on the NRPE client code.

One problem with OpenNMS comes from upgrading. If a new feature like NRPE support is added, the configuration files are often modified to provide examples of the new feature.

This causes problems when upgrading, since a change to the default file will often result in an .rpmnew file being generated, requiring a manual merge.

The existing configuration files have been streamlined to only include those features most likely to be used by end users, and the more exotic configuration options will now be shown in example files, found in the `$OPENNMS_HOME/etc/examples` directory. The default files should thus be less likely to change and require less effort during an upgrade.

2.24. Changes in OpenNMS 1.3.0

2.24.1. Bug Fixes

There have been entire sections of OpenNMS rewritten to accomodate the new features and to make the product more scalable. Please visit the [CHANGELOG](#) for a better list.

2.24.2. New Features

As with many open source projects, the OpenNMS documentation could be better. For many of the new and somewhat complex features in this development release, please refer to the Wiki page for that particular feature.

SNMP version 3 Support

Support for SNMP version 3, both Authentication and Privacy, for polling and data collection. Trap support is soon to follow. For detailed information see:

<http://www.opennms.org/wiki/index.php?page=SnmpV3Configuration>

JMX Polling and Data Collection

OpenNMS is written in Java. We like Java. But one of the shortcomings is that it is hard to determine what resources are being used within the virtual machine. The java process might be using 512 MB, but how much of that is the actual application within the machine using? Close to all of it? Very little?

The new JMX poller and data collector will allow for this information from within the virtual machine to be used in pollers and collectors. Also note this is our first non-SNMP based data collector.

More Information: <http://www.opennms.org/wiki/index.php?page=JMXConfiguration>

Serious kudos to Mike Jamison for his work on bringing this to OpenNMS.

Alarms

What happens if you get lots of the same event? It has the ability to clog up the event browser. In the office there is a Linksys router that sends an SNMP trap on every network connection (over 100,000 a day sometimes).

The new alarms system will allow important events to be "reduced" into alarms. If an event comes in with the same "reduction key" as a previous event, the alarm will increment the "count" of events, yet it will still only take up a single line in the alarm browser. Clicking on the count will bring up the event browser with just the events that have been reduced.

More Information: <http://www.opennms.org/wiki/index.php?page=Alarms>

Automations

OpenNMS provides you with a more robust solution by automating its behavior to meet the requirements of your business or organization. For example, say you have an alarm with the severity of Minor that has not been acknowledged in the last 20 minutes you might want to escalate the severity. As of version 1.2, OpenNMS has a daemon that runs SQL statements on an interval called Vacuumd. Its configuration has been enhanced with a configuration that now allows configuration of processes we're calling Automations that are defined by Triggers and Actions.

More Information: <http://www.opennms.org/wiki/index.php?page=Automations>

Roles

This is a new feature for managing Notifications. If you have, say, an On-Call role where the users change over time, this feature allows you to schedule them in advance and OpenNMS will manage that schedule for you.

More Information: <http://www.opennms.org/wiki/index.php?page=Roles>

Group Duty Schedules

Bill Ayres wrote a nifty feature called Group Duty Schedules. You add them just as you would add Duty Schedules to users. If a Group is listed as a target in a destination path, the duty schedule will apply to the whole group (individual users and roles also in the target are not affected).

The best part of this feature is that the notices are still created (and will show up on the main page). However, if a notice that is created while a group is off duty has not been acknowledged by the time they come on duty, it will be sent. So if there are, say, a group of servers that you wish to monitor but don't wish to be paged about in the middle of the night, you can use this feature so you know about them when you come back on duty the next business day.

New HTTP Monitor

The HTTP Monitor has been improved to handle virtual domains, different clients, etc. It was based in large part on work done by Erasmo Zubillaga (Bug 919).

More Information: <http://www.opennms.org/wiki/index.php?page=HttpMonitor>

JFreeChart Support

You will notice that on the front page of the OpenNMS webUI there are a number of charts displaying information from the OpenNMS database. OpenNMS now supports a JFreeChart integration. It currently only supports bar charts, and the configuration is controlled by the `chart-configuration.xml` file.

More Information: <http://www.opennms.org/wiki/index.php?page=JFreeChartSupport>

Zoom Feature for RRDs

Borrowed from Cacti, Mike Huot has created the ability to zoom in on RRD graphs within OpenNMS. Simply click on a graph to bring it up in a new window, and then drag the mouse to highlight the region you want to zoom. You can zoom in multiple times. Hit the back button on your browser to zoom out.

New Look to Availability Reports

Jonathan Sartin has made some nice changes to the Availability Reports look. Instead of bar graphs you get a calendar with colored circles representing the availability level with the actual availability number inside the circle. Just choose "Calendar" format when generating the reports to see the new style.

Chapter 3. Known Issues and Caveats

Here is the list of known issues in this release of OpenNMS.

3.1. Current Release

There are no known major issues in OpenNMS 1.6.1.

There are, however, a number of bugs and features which are slated for future 1.6 releases. They can be viewed (and voted on) in the bugzilla [1.6.2 milestone](#)

3.2. Previous Releases

- **OpenNMS 1.6.0.** A few small issues were found in the release candidates that did not make it into 1.6.0. They will be a part of the 1.6.1 release.
 - KSC reports sort by ID rather than by name (Bug [#2817](#))
 - Graphs for load averages should show typical units rather than si units (Bug [#2824](#))
 - A URL added in the discovery admin UI cannot be deleted. (Bug [#2829](#))

OpenNMS 1.5.93. OpenNMS is still not 100% compatible with PostgreSQL 8.3, because of a change in behavior regarding implicit casts. Until we can fix our code to not rely on implicit casts, a workaround is available [here](#), but it is recommended that you stick with PostgreSQL 7.4 to 8.2 for production environments until this issue is resolved.

- **OpenNMS 1.5.90 Known Issues.** Editing scheduled outages in the web UI has a number of bugs.
- **OpenNMS 1.3.10 Known Issues.** Events that reference an IP address that exists on more than one node will not be processed. Fixed in 1.3.11.
- **OpenNMS 1.3.9 Known Issues.** The User Account Self-Service window (i.e. the page that allows users to change their passwords) will cause a warning that the application is trying to close a window on IE6.
- **OpenNMS 1.3.8 Known Issues.** The order of elements in some of the XML configuration files is enforced more strictly than before. This change may cause problems with customized OpenNMS configuration files (especially event definitions) that worked fine with previous releases. Correcting the ordering of elements in these files against the XML schema definitions (available from the [wiki](#)) may be necessary.
- **OpenNMS 1.3.7 Known Issues.** The zoom feature will not work on Windows using IE or Opera. The workaround is to use Firefox. (corrected in 1.3.8)
- **OpenNMS 1.3.5 Known Issues.** Notification on the "newSuspect" event will fail. Bug [#1954](#)

Chapter 4. Supported Systems

Supported UNIX-like OSes

OpenNMS is written almost entirely in Java, and should be able to run on any system that supports the Sun Java Virtual Machine -- OpenNMS 1.3.x and higher requires Java 5 or higher. There are requirements for other programs such as PostgreSQL and Perl, but the JDK is the key requirement as most of the other packages can be compiled from source.

The following are the systems that support or are known to run OpenNMS.

Windows 2000/XP/Vista are supported as of 1.3.8.

4.1. Fully Supported

The following systems are supported out-of-the-box with native installation packages:

- **RPM-based Distributions (Using Yum).**
 - Red hat Enterprise Linux 3 and higher
 - CentOS 3 and higher
 - Fedora Core 4 and higher (including 64-bit)
 - SuSE Linux 9 and 10 (Using the Yum repository through YAST)
- **Other RPM-based Distributions.**
 - Mandriva Linux 2007 and higher (Using URPMI)
- **Debian and Ubuntu Linux.** Debian packages for Etch and later are available at the following apt repository:

```
deb http://debian.opennms.org/ unstable main
```

These same packages should work equally well with Ubuntu releases 6.10 (Edgy Eft) and higher.

- Solaris 10 (X86 and SPARC)
- **MacOSX 10.4 (Tiger), 10.5 (Leopard).** On MacOSX, the Fink distribution [packages of OpenNMS](#) are supported. See [the Fink web site](#) for more information on installing and using Fink. Also note that on MacOSX, PostgreSQL must be configured in the same manner as above for Linux. *However*, to do so you will need to update the SHM settings so that the OS allows enough resources for PostgreSQL to run with larger buffers. There are details for configuring the shared memory segments on [the Fink wiki](#).
- **Windows Server and Desktop (Windows 2000+).** Note that while it is technically possible to install on FAT32, NTFS is the only officially supported filesystem for Windows installs. Additionally, while Windows is supported, OpenNMS is much more heavily tested (and easier to maintain) on UNIX, and it is recommended that unless you have a specific reason to go with Windows, that you use one of the supported UNIX-based operating systems.

4.2. Unsupported

OpenNMS 1.3.7 and up require Java 5 (a 1.5 JDK) and PostgreSQL 7.4 or higher. In addition, for native RRD support (as opposed to the builtin Java-based JRobin round-robin database), RRDTool 1.2 is required.

Any operating system that can support these dependencies should be able to run OpenNMS. However, since many older distributions do not support packages for these applications it will be much harder to get them installed, and so they are not officially supported.

A number of distributions that used to be supported are still able to run OpenNMS but are not officially supported:

- **Gentoo.** Gentoo ebuilds used to be available but are no longer officially maintained, as we have no Gentoo packager volunteers to keep them up-to-date.
- **Red Hat Linux.** While Red hat Linux 7, 8, and 9 (and potentially even others might still work, they have long gone untested and are not recommended for production use.
- SuSE 8 and earlier
- Solaris 9 and earlier