

OpenNMS Meridian Release Notes

Copyright (c) 2015-2016 The OpenNMS Group, Inc.

OpenNMS Meridian v2016.1.2

Last updated 2017-02-02 11:42:03 -05:00

Table of Contents

| | |
|---|---|
| OpenNMS Meridian Development Team | 1 |
| OpenNMS Meridian 2016 | 1 |
| What's New in Meridian 2016 | 1 |
| Release 2016.1.0 | 5 |
| Release 2016.1.1 | 5 |
| OpenNMS Meridian 2015 | 5 |
| Release 2015.1.1 | 5 |
| Release 2015.1.0 | 5 |

OpenNMS Meridian Development Team

| |
|--|
| Tarus Balog < tarus@opennms.org > |
| David Hustace < david@opennms.org > |
| Benjamin Reed < ranger@opennms.org > |

Copyright © 2004-2016 The OpenNMS Group, Inc.

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> OpenNMS is the creation of numerous people and organizations, operating under the umbrella of the OpenNMS project. The source code is published under the GNU Affero GPL, version 3 or later and is Copyright © 2002-2016 The OpenNMS Group, Inc.

The current corporate sponsor of OpenNMS is The OpenNMS Group, which also owns the OpenNMS trademark.

Please report any omissions or corrections to this document by creating an issue at <http://issues.opennms.org>.

OpenNMS Meridian 2016

What's New in Meridian 2016

Since Meridian 2015, a number of backend changes have been made, as well as a large amount of refactoring and many new features.

Java 8

Meridian 2015 dropped support for Java 6, and now in 2016 we are dropping support for Java 7, which has been EOL'd for some time now. OpenNMS Meridian 2015 requires Java 8 to run, and we strongly recommend the latest version of Oracle JDK 8 for your platform.

PostgreSQL 9.1 or Higher

OpenNMS Meridian 2016 now requires PostgreSQL 9.1 or higher. All older versions of PostgreSQL are past their end-of-life support date.

Retired Features

- The legacy capability scanner, Capsd, has been removed from OpenNMS. It was deprecated in OpenNMS Horizon 1.12 and disabled by default in Horizon 14 and Meridian 2015 in favor of the provisioning systems of OpenNMS.
- The original link scanning daemon, Linkd, has been superseded by Enhanced Linkd since it was introduced in OpenNMS Horizon 15. Enhanced Linkd is a refactored replacement that is more efficient than the original Linkd code.
- **SVG Maps:** The SVG map feature relied on Linkd's code for drawing links between items on the map so it was also removed in OpenNMS 17.
- **Xmlrpcd:** Xmlrpcd was a daemon that relayed inventory and polling events to an external system over the XML-RPC protocol. Because you can accomplish almost all of its use cases by using the provisioning REST service, it has been removed.

- **MX4J:** The MX4J HTTP interface has been removed in favor of using more modern JMX access methods that have been added in recent versions of the JVM.

Important Upgrade Notes

- **Clean Up Events and Alarms:** Please clean up your event and alarm lists as much as possible before performing an upgrade to Meridian 2016. The upgrade process contains a large number of database schema migrations which will cause rewrites of all of the data in some tables. If you prune unused data from these tables beforehand, it will make the upgrade process much quicker.
- **Remote Poller API Change:** Due to internal API changes, the Remote Poller API has changed in Meridian 2016. If you upgrade to Meridian 2016, you will also need to upgrade all Remote Pollers attached to the system to 2016 as well.
- **RMI Security Updates:** The RMI registry has been changed to only listen on localhost by default. It also now requires authentication using an OpenNMS administrator account.
- **JMX Updates:** JMX is no longer enabled by default on port 18980. If you have reason to access your OpenNMS instance remotely over JMX, you can reenabling it by adding `-Dcom.sun.management.jmxremote.port=18980` and `-Dcom.sun.management.jmxremote.ssl=false` to `ADDITIONAL_MANAGER_OPTIONS` in `$OPENNMS_HOME/etc/opennms.conf`. Complete documentation on enabling remote access and configuring authentication is available in the admin guide. Note that OpenNMS-JVM monitoring of the local instance (ie, localhost) will still work by using the JMX Access API, without needing to enable the TCP port.

API Changes

Several classes changed location in Meridian 2016 and these changes may require you to update configuration files or scripts with the new names.

- The `IndexStorageStrategy` and `PersistAllSelectorStrategy` classes moved into the `org.opennms.netmgt.collection.support` Java package. These classes are heavily referenced in data collection configuration files. There is an upgrade task that should update all data collection files when you run the `install` command after upgrading OpenNMS.
- All `EventUtils` class methods were consolidated inside the `org.opennms.netmgt.model.events.EventUtils` class.
- Monitoring locations for the Remote Poller are now viewed and configured in the UI under `/locations/index.jsp`. Monitoring locations were previously configured in the `etc/monitoring-locations.xml` file but are now stored in `monitoringlocations` database table.
- All data previously in the `location_monitors` table is now in the `monitoringsystems` table.
- The `ifIndex` column has been removed from the `ipInterface` table. To fetch the `ifIndex`, you need to join to the `snmpInterface` table by `nodeId`.
- The `nodeId` and `ipAddr` columns have been removed from the `ifServices` table. To fetch these columns, you need to join to the `ipInterface` table.
- The `nodeId`, `ipAddr` and `serviceId` columns have been removed from the `outages` table. To fetch these columns, you need to join to the `ifServices` table.

New Features

- There is a new REST service that can be used to export performance data. This service can be used to easily export OpenNMS metrics into external graphing engines such as Grafana. See the REST API documentation for details.

- **Newts Storage for Data Collection:** In addition to RRD and JRobin, OpenNMS 17 data collection now supports our [Newts](<http://opennms.github.io/newts/>) time-series data store that uses the highly-scalable Apache Cassandra database as the storage backend. This storage solution provides higher performance than the file-based RRD and JRobin databases and practically unlimited scalability when using a cluster of Cassandra nodes. You can drastically reduce CPU and storage load on your OpenNMS system and increase the number of metrics stored per second by using the Newts storage engine.
- **Measurements REST API Filtering:** The measurements REST API now supports the use of arithmetic filters on the returned datasets. This allows you to perform calculations or statistical operations on the data via server-side processing.
- **Grafana Panel:** An optional panel has been added to the UI to allow you to integrate links to Grafana graphs in the UI.
- **Streamlined UI Style:** The style of the UI has been refined to provide a clearer, more modern look-and-feel.
- **Alarm Heatmap:** A heatmap visualization has been added that lets you quickly visualize alarm status and outages within a category or on a node.
- **FreeIPA Kerberos Authentication:** We've added a sample configuration for FreeIPA Kerberos SSO to our `jetty-webapps/opennms/WEB-INF/spring-security.d` directory.
- **JMX Configuration UI and CLI:** The JMX Configuration web UI and CLI have been rewritten to make it easier to generate complex JMX data collection configurations.
- **JMS Alarm Northbounder:** The implementation of a JMS northbounder for sending OpenNMS Alarms to external JMS listeners has been completed. Thanks to David Schlenk (@dschlenk) for this contribution!
- **OTRS 3.1+ Ticketing Integration:** Our OTRS ticketing integration has been updated to work with the API in OTRS 3.1 and higher. Thanks to Jonathan Sartin (@blackpoplar) for this contribution!
- **Easier Remote Poller Configuration:** Monitoring locations can now be associated with multiple polling and collection packages. This can make some Remote Poller scenarios easier to configure.

Events

New or updated trap definitions have been added for the following classes of devices:

- A10 AX Load Balancer
- Avocent DSView
- Citrix NetScaler
- Dell Force 10
- Evertz 7800FR Multiframe
- 7880-IP-ASI-IP
- 7881-DEC-MP2
- 7780-ASI-IP2
- Infoblox

- Mikrotik RouterOS
- ~50 new NetApp Events
- OpenSSH syslog events
- Postfix syslog events
- Procmail syslog events
- Raytheon NXU-2A
- Siemens HiPath
- Veeam Backup/Replication

Data Collection

- Fortinet Fortigate
- Sonicwall

Core Updates

As always, many updates and cleanups have been made to the OpenNMS core, through refactoring, addition of unit tests, and other code modernization.

- A few subsystems have been updated to run properly under OSGi.
- A large number of places where we were still using JDBC calls have been converted to use our DAO infrastructure and Hibernate.
- The OpenNMS ReST APIs now support CORS properly to aid in writing web applications that consume or update OpenNMS data.
- The JMX detector, monitor, and collector were refactored for modularity.
- The Dashboard was rewritten using the Vaadin UI toolkit to improve and modernize its look-and-feel.
- Bean Scripting Framework (BSF) notifications and the BSFMonitor were optimized and are now much more efficient. (Thanks to David Schlenk for this contribution!)
- RTC, which calculates the availability percentages for the category panel on the main page, was rewritten using Spring for initialization and using database calls to perform availability calculations. This will improve its maintainability.
- The web controllers for provisioning, RANCID integration, reports, the node list, and Remote Poller administration were rewritten to modernize their code.
- The REST portion of the OpenNMS webapp was modularized into its own project.
- The REST interface has been refactored to be based on Apache CXF 3.1.4.
- The web UI service layer was separated from the main web UI to improve modularity.
- The Jasper and statsd reports have been changed so that they use a new unified interface for fetching data from either

RRD, JRobin, or Newts.

- The Discovery engine has been refactored into its own project.
- The Dashboard has been rewritten using the Vaadin toolkit to modernize its look-and-feel.
- The `opennms-assemblies/jmx-config-generator-onejar` project has been moved to `features/jmx-config-generator`.

Dependency Updates

A number of internal libraries have been upgraded for bugfixes and new features. None of these updates should require configuration changes.

- Spring has been upgraded from 3.2.9 to 4.0.5.
- Spring Security has been upgraded from 3.1.7 to 3.2.7.
- Drools has been upgraded from 5.1.1 to 6.0.1.
- Apache Camel has been upgraded from 2.13.2 to 2.14.1.
- Smack (used for XMPP notifications) has been upgraded from 3.0.4 to 4.0.6.
- The webapp schemas have all been updated to the Servlet 3.0 specification.
- Jasper Reports has been upgraded from 5.6.1 to 6.1.1.

Release 2016.1.0

Release 2016.1.0 is the first release of OpenNMS Meridian 2016. It is based on OpenNMS Horizon 17.1.1, plus a number of smaller bug fixes.

The codename for 2016.1.0 is *Mercator*.

Release 2016.1.1

Release 2016.1.1 is an update to 2016.1.0 that provides a fix for upgrades.

The codename for 2016.1.0 is *Cassini*.

OpenNMS Meridian 2015

Release 2015.1.1

Release 2015.1.1 is the second release of OpenNMS Meridian 2015. It is based on OpenNMS Horizon 14.0.3, the Bootstrap GUI shipped in OpenNMS Horizon 15, and many smaller bug fixes.

The codename for 2015.1.1 is *London*.

Release 2015.1.0

Release 2015.1.0 is the first release of OpenNMS Meridian 2015. It is based on OpenNMS Horizon 14.0.3, the Bootstrap GUI shipped in OpenNMS Horizon 15, and many smaller bug fixes.

The codename for 2015.1.0 is *Greenwich*.