
Release Notes for Zenoss

Service Dynamics Version 4.1.1

Copyright © 2011 Zenoss, Inc., All rights reserved. Redistribution or duplication of any portion of this document is prohibited without the express written consent of Zenoss, Inc.

Zenoss and the Zenoss logo are trademarks or registered trademarks of Zenoss, Inc. in the United States and other countries. All other trademarks, logos, and service marks are the property of Zenoss or other third parties. Use of these marks is prohibited without the express written consent of Zenoss, Inc. or the third-party owner.

Cisco Unified Computing System is a trademark of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

Flash is a registered trademark of Adobe Systems Incorporated.

Linux is a registered trademark of Linus Torvalds.

Oracle, the Oracle logo, MySQL, and Java are registered trademarks of the Oracle Corporation and/or its affiliates.

SNMP Informant is a trademark of Garth K. Williams (Informant Systems, Inc.).

Sybase is a registered trademark of Sybase, Inc.

Tomcat is a trademark of the Apache Software Foundation.

ESX, RabbitMQ, vCloud, VMware and vSphere are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions.

Windows is a registered trademark of Microsoft Corporation in the United States and other countries.

All other companies and products mentioned are trademarks and property of their respective owners.

Part Number: 23-122011-4.1-v02

| | |
|---|----|
| 1. About These Notes | 2 |
| 1.1. What's New? | 2 |
| 1.1.1. zentune | 2 |
| 1.1.2. zenwebserver | 3 |
| 1.1.3. ZenHub Configuration Options | 3 |
| 1.1.4. Invalidation Workers (ZenHub) | 3 |
| 1.1.5. Audit Logging | 4 |
| 1.1.6. Nexus7K ZenPack | 4 |
| 1.1.7. zenping | 4 |
| 1.1.8. Additional Enhancements | 5 |
| 1.2. Downloading Service Dynamics | 5 |
| 1.3. Supported Software and Environments | 5 |
| 1.4. Resource Management | 6 |
| 1.4.1. Installation, Upgrade and Implementation Notes | 6 |
| 1.4.2. Known Issues | 9 |
| 1.5. Global Operations Management | 10 |
| 1.5.1. New Features | 10 |
| 1.5.2. Defect Fixes | 10 |
| 1.6. Documentation Changes and Additions | 11 |
| 1.6.1. Resource Management Administration | 11 |

| | |
|--|----|
| 1.6.2. Resource Management Extended Monitoring | 11 |
| 1.6.3. Zenoss Service Dynamics Impact and Event Management | 12 |
| 1.7. Reporting Problems and Providing Feedback | 13 |
| 1.7.1. Product Documentation Feedback | 13 |
| 1.8. Defects Fixed in this Release | 13 |

1. About These Notes

These release notes contain important information about this release of Zenoss Service Dynamics, including:

- New features
- Where to download the software
- Supported software and environments
- Update information, known issues, and documentation changes (by product component)
- Reporting problems and providing feedback
- Defects and issues fixed in the release

Service Dynamics Version 4.1.1 includes these product component versions:

| Component | Version |
|------------------------------|---------|
| Analytics and Optimization | 4.1 |
| Global Operations Management | 1.0.2 |
| Impact and Event Management | 4.0 |
| Resource Management | 4.1.1 |

1.1. What's New?

This version of Service Dynamics offers new features and improvements in the following areas.

1.1.1. zentune

ZenPacks.zenoss.AutoTune ("zentune") is a script that analyzes your system configuration and makes recommendations for better performance.

Usage

To run the zentune script, use this command:

```
$ zentune run
```

The script prints the current and optimal values for several configuration parameters. Recommendations for configuration changes are printed at the end of the report.

Sample Report

```
----- ZOPE -----
[OK] Object cache: 50000 (250% of suggested value 20000)
[OK] Pool size: Current value 7 (no greater than 10 recommended)
[OK] RelStorage cache: Not set
[!!] Cache servers: 0 of 1 servers available
    127.0.0.1:11211 is inaccessible
[OK] Maximum number of session objects: 100000 (100000 recommended)
[OK] Debug mode: off
[!!] Check interval: Current value 500 (suggested value 1493)
[OK] Application server: 1 Zope process for 0 users
[OK] Application server: 4 threads per Zope instance

----- HUBS -----
[OK] Hub: localhost: 1 collector
```

```
----- GLOBAL -----
[--] Global config sip size: not set
[--] Global config sip delay: not set
[--] Event flush chunk size: not set
[--] Maximum queue length: not set

----- EVENTS -----
[!!] zeneventd object cache: Current value 1000 (20% of suggested value)
[OK] zeneventd workers: 2 workers

----- RESOURCES -----
[!!] Processes: 2 cores for
    2 zenhub workers
    2 zeneventd workers
    1 Zope process
    1 zeneventserver
[OK] Memory: 4141654016 bytes of total memory

----- RECOMMENDATIONS -----
* Make sure memcached is running
* Set python-check-interval in zope.conf to 1493
* Add 'configsipsize 50' line to the global.conf
* Add 'configsipdelay 5' line to the global.conf
* Add 'eventflushchunksize 2000' line to the global.conf
* Add 'maxqueuelen 20000' line to the global.conf
* Set cachesize in zeneventd.conf to at least 5000
* Number of CPU-intensive processes exceeds available cores.
  Consider moving to a distributed zenhub and/or running zeneventd
  on another machine.
```

1.1.2. zenwebserver

The zenwebserver ZenPack deploys and manages multiple Zope instances. Refer to the zenwebserver documentation (appended to these Notes) for complete information.

1.1.3. ZenHub Configuration Options

New options apply to all collector daemons and control how those daemons request configurations from ZenHub:

- **configsipsize** -- If set to a non-zero value, the daemon requests n device configurations from ZenHub (where n is the value for configsipsize). The default value is 25.

By requesting device configurations in batches, the option allows the daemon to start monitoring devices as the device configuration is loaded. Also, the smaller batches prevent ZenHub from locking up a process for long periods of time.

If set to a value of 0, then all configurations are downloaded at once; monitoring does not commence until all configurations have been loaded by the collector. On systems with a large number of devices, the collector may be waiting a long time to download all the configurations.

- **configsipdelay** -- Controls how many seconds (at most) to wait between making device configuration requests. This option is ignored if the value of configsipsize is 0. The default value is 1.

```
--configsipsize=CONFIGSIPSIZE
    Max number of device configurations to load at once,
    default 25 (0 == all devices)

--configsipdelay=CONFIGSIPDELAY
    Delay in seconds between device configurations
    loading, default 1
```

1.1.4. Invalidation Workers (ZenHub)

This addition to ZenHub allows it to process configuration updates in parallel workers, freeing up the hub for other work and permitting quicker delivery of updates to the collectors. Relevant ZenHub configuration options are:

- **invalidationworkers** -- Specifies the number of invalidation workers that should process configuration changes. The default value is 1. May need to be raised if your database comprises more than 5000 devices, or if your configuration experiences a high rate of change (such as frequent addition or removal of templates, or frequent configuration property changes).

To determine if you need to increase the number of invalidation workers, run the following command:

```
$ sudo rabbitmqctl list_queues -p /zenoss
```

Look at the number of messages in the "invalidations" queues. If these are greater than 0 for any length of time, then consider adding invalidation workers.

- **invalidationchunksize** -- Specifies the number of changes that will be sent to the workers at one time. The default value is 100. This value likely does not need to be changed.
- **invalidationlimit** -- Specifies the number of changes each worker will process before it shuts down and is replaced by a new worker. The default value is 5000. This value likely does not need to be changed.
- **identifier** -- Specifies the name by which the hub will be known. This allows workers to communicate with the hub by using specific queues. The default value is *HubName:HubPort*. This value does not need to be changed.

Note

This feature requires ZenPacks.zenoss.EnterpriseCollector-1.1.4.

1.1.5. Audit Logging

Enhancements to the audit log feature, which maintains logged information in a format optimized for searching and reporting. Along with various defect fixes, these include:

- Extended audit capabilities, including auditing of logouts and VMware
- Command that can be run by users and shell scripts to audit actions:

```
zensendaudit 'Running script zenfixcatalog to fully reindex the catalog'
```

- Python scripts and zendmd can do this also:

```
from Products.ZenMessaging.audit import auditComment
auditComment("Manually fixed the catalog in zendmd")
```

1.1.6. Nexus7K ZenPack

Enhancements to the Nexus7K ZenPack, including:

- Modeling where only the default VDC is enabled
- Modeling using NX-OS firmware 5.2(1)
- Modeling and monitoring of loopback interfaces
- Corrections to modeling of MTU size of network interfaces and interface ranges for VLANs.

1.1.7. zenping

In Versions 4.0.x and 4.1, zenping introduced event suppression, which occurs when a resource is detected as down, and another resource (along its traceroute, or gateway) is also down. This allows users to focus efforts on the primary cause of a ping down event.

In Version 4.1.1, zenping:

- Relies on a subprocess (nmap) to handle batch pinging devices. Time-critical tasks are no longer handled in-process.
- Uses nmap to provide traceroute information. nmap implements an efficient "reverse traceroute" method that allows zenping to capture the network topology quickly and frequently.
- Will traceroute every `--traceroute-interval pingCycles` (1 time every 5 minutes).

In Version 4.1.1, zenping does not support:

- Multiple routes to a host. This is not problematic, as suppressed events are sent only when the remote resource is offline.
- IPv6 for traceroute. In this case, zenping defaults to ping6 to determine availability.
- Event suppression for Ipv6 addresses.

The PING data source now includes AVG, MIN, MAX, STDDEV round trip times, as well as the percentage of lost packets.

1.1.8. Additional Enhancements

Other features and improvements include:

- Significant improvements to the event processing system, increasing to 100M+ events per day.
- Advanced user configuration options in the Resource Manager interface (Advanced > Settings). They control how data loads, how much data is loaded, and filter and search configuration.
- Improvements to published JSON API examples, including the introduction of a Python client library. Browse to: http://community.zenoss.org/community/documentation/official_documentation/api access this update.
- Resource Manager now stores session information in a persistent database, which is created in MySQL automatically. Zope's configuration file is automatically altered to use persistent sessions, if possible.
- HP-UX ZenPack updates, including stability improvements and support for Itanium processors.
- Queuing mechanism that allows separation of event processing, horizontal scaling, and the ability to subscribe to event queues for custom processing.
- REST API for querying events.
- Triggers and notifications, which offer more robust and granular options for event alerting rules.
- Zenoss DataStore, an updated database that provides partitioning to support the new Resource Manager event processing system.
- RelStorage, which improves the performance of Zope object database queries.
- Support for SNMPv3 traps.
- Ability to monitor devices directly over IPv6.
- All performance collecting daemons have moved to the collector framework for more consistent operation and higher reliability.
- Can graph ping performance information for devices and interfaces. Per-interface pinging requires a component template be created for the interface to be monitored.
- Incremental device loading per daemon. This enables monitoring to start before all device information is loaded, and helps spread the monitoring load over a period of time rather than all at once.

1.2. Downloading Service Dynamics

Service Dynamics is available from the Zenoss Support Portal, at:

<https://support.zenoss.com>

Contact your Zenoss representative for more information.

1.3. Supported Software and Environments

You can install this version of Service Dynamics on these Linux® platforms:

- RedHat® Enterprise Linux 5
- CentOS 5 (verified with CentOS 5.7)

For each system that will access Service Dynamics through a Web browser, you need:

- Adobe® Flash® Player 10 or later version
- One of these browser versions:
 - Firefox (verified with version 8)
 - Internet Explorer (verified with versions 7, 8, 9)
 - Chrome (verified with version 15)

The following partial list of resources can be managed by Resource Manager:

- Cisco Unified Computing System™
- VMware ESX® Infrastructure, VMware vSphere™, VMware vCloud™
- NetApp®
- Windows Server (2000, 2003, 2008), Windows XP, and Windows Vista®, Windows 7
- Linux or other UNIX® server
- OpenStack
- Tomcat™ and other Java®/JMX servers
- Any SNMP- or SSH-enabled device

1.4. Resource Management

Read the following sections for information related to Service Dynamics Resource Management.

1.4.1. Installation, Upgrade and Implementation Notes

Read this section for important installation, upgrade, and implementation updates. For complete installation and update instructions, refer to the latest versions of the installation documentation.

1.4.1.1. Upgrade

Zenoss recommends that you upgrade to the current release to receive the latest features and defect fixes. Refer to the following table to determine the upgrade path you must follow when upgrading to a new version of Resource Manager.

| If your current version is: | You can upgrade directly to this version: |
|-----------------------------|---|
| Zenoss Enterprise 3.2.x | Resource Manager 4.1.1 |
| Resource Manager 4.0.2 | Resource Manager 4.1.1 |
| Resource Manager 4.1.0 | Resource Manager 4.1.1 |

Table 1. Upgrade Paths

- First, install any missing prerequisites, as outlined in the chapter titled "Installing for RHEL 5 or CentOS 5" in *Resource Management Installation*. Then follow the instructions in the chapter titled "Upgrading" to upgrade your Zenoss Enterprise 3.2.x, Resource Manager 4.0.2, or Resource Manager 4.1.0 instance to Resource Manager 4.1.1.
- If you are running Resource Manager 4.0.2, and have Impact installed, contact Zenoss Support before upgrading to Resource Manager 4.1.1.
- Zenoss Resource Manager includes a fully redesigned event processing and storage system. At this time, upgrades to Resource Manager will not automatically migrate events from Zenoss Enterprise 3.2.x. A new, empty event table is created in the new schema after upgrade.

If you want to migrate events to Resource Manager as part of your upgrade, then a managed migration path is available. Zenoss recommends you contact Zenoss Professional Services for assistance, and then create an output file of your current MySQL events database.

To create the output file, enter this command:

```
mysqldump -u root -p [ROOT_PASSWORD] events | gzip -c > zenoss_events.sql.gz
```

- If you are running with a load-balanced configuration with multiple Zope instances, or have a highly customized distributed collector environment, your configuration may be slightly modified during upgrade to this release of Resource Manager. Specifically, if your main Zenoss 3.2.x master has a `DAEMONS_TXT_ONLY` and `daemons.txt` file to restrict the daemons run on your master server, you will be affected. The upgrade process will save these two files to your `$ZENHOME/etc` directory with the extension "rpmsave," and your post-upgraded system will return to the default daemon starting behavior.

After upgrade, you can safely modify these files and restrict the daemons started on your master server again. If you do so, please take notice of the daemons needed for Resource Manager Version 4.1.1, as these have changed from earlier releases.

- Zenoss does not test or evaluate community, customer-developed, or other custom ZenPacks for upgrades.

If you are using one or more ZenPacks that are not installed through the standard Resource Manager installation process, you should contact the ZenPack author about its compatibility with this release. Do not upgrade until you ensure compatibility of all custom ZenPacks. Zenoss further recommends you test the ZenPack for upgrade compatibility in a test environment.

1.4.1.2. Updating Custom ZenPacks

- Custom ZenPacks must be compatible with Python 2.7. You must upgrade custom ZenPacks with Python 2.7-compiled extensions.
- If you have a custom ZenPack running on Zenoss Enterprise 3.2.x that includes alerting rules with custom actions, then you must modify the ZenPack before you can use it with Resource Manager. Otherwise, the system will produce warning messages similar to this:

```
WARNING:zen.migrate: <rule>: Successfully migrated rule to Trigger, \
but was unable to create a Notification - rule has invalid or unknown action type: <action>
```

You must add the following code to your custom ZenPack, in `configure.zcml`:

```
<configure xmlns="http://namespaces.zope.org/zope">
  <utility factory=".actions.MyAction"
            provides="Products.ZenModel.interfaces.IAction"
            name="my_action"/>
</configure>
```

Read the `IAction` class declaration in `$ZENHOME/Products/ZenModel/interfaces.py` for documentation of the attributes and methods your action needs to implement. Add the following code to `actions.py` in your custom ZenPack:

```
class MyAction(IActionBase):
    implements(IAction)

    id = 'my_action'
    name = 'My Action'
    actionContentInfo = ICommandActionContentInfo

    def execute(self, notification, signal):
        ...

    def updateContent(self, content=None, data=None):
        ...
```

Note that the name attribute in the `configure.zcml` file matches up to the `id` attribute in the `actions.py` file. Examples of implementing actions are in `$ZENHOME/Products/ZenModel/actions.py`, which declares four core actions: email, page, command, and trap.

1.4.1.3. Event Console Filtering

Event console filtering behavior in Resource Manager is different than that of Zenoss 3.2.x, as follows:

- Regular expressions are not supported when filtering by device or component.

- In device/component fields, if you enclose a query in double quotes, it searches for that device or component by using an exact match search (fastest). For example, searching for "abc" matches abc, but not abcd or babc.

For more information about event console filtering, see the chapter titled "Using Resource Manager" in *Resource Management Administration*.

1.4.1.4. Zenoss DataStore

- Command-line access to the Zenoss DataStore is available only to the zenoss user; Zenoss DataStore tools are located only in the zenoss user's \$PATH.

Run all Zenoss DataStore commands as the zenoss user, as in:

```
su - zenoss
zends -u root
```

- If you currently are running Zenoss DataStore 5.5.13, you must install an updated version (Version 5.5.15). To install the new version, run this series of commands:

```
service zenoss stop
service zends stop
rpm -Uvh zends-5.5.15-1.Version.el5.i386.rpm
service zends start
service zenoss start
```

- When upgrading, `zends.cnf` file customizations may need to be manually merged. For more information, refer to the chapter titled "Upgrading" in *Resource Management Installation*.
- When upgrading with a remote Zenoss DataStore configured, any modifications made to `$ZENHOME/bin/zenoss_init_pre` OR `$ZENHOME/bin/zenoss_upgrade_pre` will be preserved automatically as part of the upgrade. Modified versions will be backed up to `zenoss_init_pre.rpmsave` and `zenoss_upgrade_pre.rpmsave`, respectively. You must merge any changes to these settings into the `zenoss_init_pre` and `zenoss_upgrade_pre` files before starting Zenoss for the first time. (Internal Defects 29584, 29553)

1.4.1.5. IPv6

Resource Manager can model and monitor IPv6-addressed devices by using SNMP, Telnet or SSH. Ping monitoring is also supported for IPv6. Resource Manager installation (and communications links among Resource Manager components) must continue to be over IPv4. This includes all links between local or distributed ZenHubs and collectors, and Resource Manager dependencies (such as RabbitMQ).

When adding a new device to Resource Manager, the DNS resolution of the device name dictates whether Resource Manager attempts to connect by using IPv4 or IPv6. If you enter an IP address directly (either as a device name or by manually changing a device's management IP), then Resource Manager can be forced to use IPv4 or IPv6 manually.

If you want to monitor devices in your infrastructure that have IPv6 addresses, make sure you have installed and configured an IPv6 interface on your Resource Manager server.

1.4.1.6. ZenHub

This release of Resource Manager uses RelStorage, a backend for ZODB that stores pickles in a relational database. With the switch to RelStorage, the `pcachesize` and `pcachedir` options are no longer relevant. Remove these options from any daemon configuration (`.conf`) files. (Defect 28620)

1.4.1.7. TrapForwarder ZenPack

The TrapForwarder ZenPack is deprecated in this release. SNMP traps are now handled as part of notifications. For more information, see the chapter titled "Using Resource Manager" in *Resource Management Administration*.

1.4.1.8. Reports

The process for determining device state has changed for the Availability report. Previously, if an event was reported that indicated a device down condition before the start of the report window, this event would not be

seen, and the device would be considered 100% available. The Availability report now looks for all open events reported before or during the reporting window, and computes availability by using the device state based on prior open events.

1.4.1.9. Event Commands and Alerting Rules

Do not escape event command messages and event summaries. For example, write this command as: `{evt/summary}` (rather than `echo '$evt/summary'`). (Internal Defect 28755)

Alerting rules or event commands that perform starts or ends with queries on the `ntevId` field, or perform comparisons against a non-numeric `ntevId` field, cannot be migrated to the equivalent 4.0.x trigger rule. These alerting rules or event commands must be manually migrated to the appropriate trigger after installation. (Internal Defect 28818)

1.4.1.10. Debug Logging from `zeneventd`

To enable debug logging for `zeneventd`, you must run `zeneventd` with the `-v10` option. (Internal Defect 28626)

```
zeneventd start -v10
```

1.4.2. Known Issues

The following issues are known for this version of Resource Manager.

1.4.2.1. Distributed Collector

Only one `zentrapp` instance can be run on a server, as it must bind to the SNMP trap port (162). If you install multiple collectors on the same server, you must assign different port numbers to additional `zentrapp` daemons. Attempting to run additional `zentrapp` daemons using the same port will cause them to fail at startup. (Defect 29157)

1.4.2.2. Active Directory

When monitoring Windows 2008 SP1 servers using the ActiveDirectory ZenPack, some performance counters expected by the ZenPack will not be available, and will generate error messages and cause missing performance counters. See the section "Changes to performance counters" at [http://technet.microsoft.com/en-us/library/cc754463\(WS.10\).aspx](http://technet.microsoft.com/en-us/library/cc754463(WS.10).aspx) for more details about changes in Windows 2008. (Defect 28640)

1.4.2.3. ZenPacks

- The definitions of the Text and TextLine schema have changed. Any ZenPacks that use Text or TextLine in their interface definitions must be updated. Text is now TextLine, and TextLine is now Text. An example about how to preserve compatibility with 3.x and 4.x ZenPacks is available in the ZenPackTemplate GitHub repository (<https://github.com/zenoss/ZenPackTemplate/blob/master/CONTENT/interfaces.py>). (Defect 28507)
- The 1.4.0 version of the CiscoUCS ZenPack in Resource Manager changes the event class structure for events that originate from a UCS Manager. The previous event classes `/CiscoUCS/Events` and `/CiscoUCS/Faults` are no longer present. Events that come in will be in the `/Unknown` class unless they are mapped to the `/Status/Blade` or `/Status/Chassis` class. If you have created custom mappings or transforms for the `/CiscoUCS/Events` or `/CiscoUCS/Faults` event classes, they will be lost.
- The `zope.app` namespace was deprecated by Zope and has been removed. Resource Manager includes a backwards compatibility layer, but any references in ZenPacks to `zope.app.*` should be changed to `zope.*` instead. (Defect 28930)

1.4.2.4. Site Window Portlet

Some Web sites may not be compatible with the Site Window (Welcome) portlet that appears on the Dashboard. Before customizing this portlet to point to another Web site, make sure that site is not running a JavaScript "frame breaker" script. (Defect 27151)

1.4.2.5. Internet Explorer 9

When using Internet Explorer 9, Resource Manager user interface page elements may not load and display correctly. To work around this issue, go to the Internet Options **Advanced** tab (from Settings > Internet Options), and then reset Internet Explorer settings.

1.4.2.6. Reports

After upgrade, reports may not display correctly for users with no assigned roles. (Internal Defect 28822)

To work around this issue, re-index the catalog with the command:

```
zencatalog --reindex
```

1.4.2.7. Windows Services

Resource Manager displays a status of "unknown" for Windows services that are disabled or not monitored. (Defect 27028)

1.4.2.8. Zenoss Global Dashboard

Zenoss Global Dashboard is not supported in this release. If you are running Zenoss Global Dashboard, you must contact Support before upgrading.

1.4.2.9. Collector Performance Graphs

Collector default performance graphs for cycle times show NaN values for zenping, zenperfsnmp, and zenstatus daemons. (Internal Defect 29378)

1.5. Global Operations Management

This section lists updates and fixes for Version 1.0.2 of Service Dynamics Global Operations Management.

1.5.1. New Features

In this version, the HTTP client used to propagate actions to local and remote Resource Manager instances has been rewritten for performance and reliability (using asynchronous I/O instead of threading).

1.5.2. Defect Fixes

Fixes in this version of Global Operations Management are:

- Properly timeout HTTP connections to local / remote Zenoss instances. Previously, zengomd could fail to shutdown if a thread was blocked waiting for a response from a HTTP request.
- Removed an unnecessary logged exception in event.log when a target queue didn't exist on a GOM source system.
- Fix GOM criteria filter to properly support filtering events based on the event count.
- Fix GOM criteria filter to properly handle 'contains' criteria on device organizers (Groups/Systems).
- Do not fail if a note cannot be propagated to an event on a source or target system if the event no longer exists.
- Fix error when 'Update Model' preference is enabled for a source and an event class doesn't exist on the target system.
- Avoid unnecessary object creation if 'Update Model' preference is enabled.
- Work around ConflictError when creating model elements when the 'Update Model' preference is enabled.
- Optimize the query used to search for existing devices when creating model elements.
- Set the title of created devices to the title from the forwarded event if 'Update Model' is enabled.

- Properly tag forwarded events with the UUID of the device when the 'Update Model' preference is enabled. This enables viewing events for the device / device class in the event console for the device / device class.
- Fix errors performing 'Update Model' operations after the hub has been restarted.

1.6. Documentation Changes and Additions

The following sections detail changes and additions to the latest published product documentation.

1.6.1. Resource Management Administration

- In the chapter titled "Using Resource Manager," the following section has been added:

Advanced User Interface Configuration

To access advanced user interface configuration options, select **Advanced > Settings**, and then select **User Interface** in the left panel. This area lets you configure options such as how data loads, how much data is loaded, and filter and search options.

Select options or enter information:

- **Enable Live Search** - Disable this option to turn off the live search feature. By default, live search is enabled.
- **Enable Incremental Tree Loading on the Infrastructure Page** - Enable this option to load the infrastructure tree one node at a time. If disabled (the default), then the infrastructure tree is loaded all at once. You might enable this option if you have a complex hierarchy of organizers and device classes and want to improve your UI load response time.
- **Enable Tree Filters** - If disabled, then tree filters (the text input area that allows you to filter the information displayed) are hidden on all pages. This option is enabled by default.
- **Device Grid Buffer Size** - Specify the number of device data rows to fetch from the server for each buffer request. The default buffer size is 100 rows.
- **Device Grid Near Limit** - Specify the limit at which device data is fetched from the server, based on the buffer size. The default limit is 20 rows.
- **Component Grid Buffer Size** - Specify the number of component data rows to fetch from the server for each buffer request. The default buffer size is 50 rows.
- **Component Grid Near Limit** - Specify the limit at which component data is fetched from the server, based on the buffer size. The default limit is 10 rows.
- **Event Console Buffer Size** - Specify the number of event rows to fetch from the server for each buffer request. The default buffer size is 200 rows.
- **Event Console Near Limit** - Specify the limit at which component data is fetched from the server, based on the buffer size. The default limit is 40 rows.

When complete, click **Save** to save your changes.

- In the chapter titled "Working with Devices," the section titled "Modifications" has been removed. Device modifications information is now available in the audit log.
- In the chapter titled "Properties and Templates," the following device configuration property has been added:
 - `zInterfaceMapIgnoreDescriptions` -- (string) Filters out interfaces based on description.
- The location of the Zope Page Templates Reference listed in the Appendix titled "TALES Expressions" is incorrect. This reference can now be found at:

<http://docs.zope.org/zope2/zope2book/AppendixC.html>

1.6.2. Resource Management Extended Monitoring

- In the chapter titled Hewlett Packard Unix:
 - The Prerequisites information has changed:

| Prerequisite | Restriction |
|--------------------------|--|
| Product | Resource Manager 4.x, Zenoss 2.5 or higher |
| Required ZenPacks | ZenPacks.zenoss.HpuxMonitor |
| Supported HP-UX Releases | HP-UX 11 |
| Supported Processors | PA-RISC, Itanium |

Table 2. HP-UX Prerequisites

- The section titled "Limitations" has been removed.
- In the chapter titled LDAP Authentication, the following note was removed:

Zenoss recommends that you make sure that your LDAP server requires at least four successive failures to lock an account. Due to authentication design, each login to Resource Manager goes through three different Web pages. Each one of these pages requests a user authentication, which ends up making a single call to the LDAP backend. Thus, if the user makes one mistake and the LDAP server locks the account on three successive failures, the user's account will be locked even though he specified the password once.

- In the chapter titled Multi-Realm IP Networks, the following prerequisite was added:

Before setting up multi-realms, you must delete all Resource Manager networks. (These are automatically recreated.)

- In the chapter titled Nexux7k, the following limitation has been added:

To model Cisco Nexus 7000 series switches with multiple enabled VDCs, the user name used to perform modeling must be authorized to run the following command on the switch:

```
show running-config vdc-all
```

This command requires authenticating as a user with one of these user roles:

- network-admin
- network-operator
- vdc-admin
- vdc-operator

1.6.3. Zenoss Service Dynamics Impact and Event Management

In the chapter titled "Working with Impact," a new section was added.

1.6.3.1. Adding Context Information to Notifications

In addition to the context information available to notifications (see the section titled "Working with Notifications" in *Resource Manager Administration*), Impact events populate a variable (`esa`) which has one variable (`causes`), which is a list. Each list entry is an event, with these attributes:

- `evt`
- `clearEvt`
- `eventSummary`
- `clearEventSummary`
- `urls`

Another entry, `impactChain`, contains the chain of resources that Resource Manager uses to determine that this event caused the service problem. The causes are listed in order of probability of root cause, from highest to least.

To iterate over the list items in `causes` using TALES, you can use the `tal:repeat` clause. For example:

```
<tal:block tal:repeat="item esa/causes">
Impact Chain: ${item/impactChain}
Device: ${item/evt/device}
Component: ${item/evt/component}
Severity: ${item/evt/severity}
Time: ${item/evt/lastTime}
Message:
${item/evt/message}
<a tal:attributes="href item/urls/eventUrl">Event Detail</a>
<a tal:attributes="href item/urls/ackUrl">Acknowledge</a>
<a tal:attributes="href item/urls/closeUrl">Close</a>
<a tal:attributes="href item/urls/eventsUrl">Device Events</a>
</tal:block>
```

To use solely the most probably cause, use `tal:define` and a Python expression:

```
<tal:block tal:define="topcause python:esa['causes'][0]">
Impact Chain: ${topcause/impactChain}
Device: ${topcause/evt/device}
Component: ${topcause/evt/component}
Severity: ${topcause/evt/severity}
Time: ${topcause/evt/lastTime}
Message:
${topcause/evt/message}
</tal:block>
```

1.7. Reporting Problems and Providing Feedback

To contact Zenoss Customer Support, go to the support portal at:

<https://support.zenoss.com>

1.7.1. Product Documentation Feedback

Zenoss welcomes your comments and suggestions to help us improve our product documentation. Please send your comments to:

docs@zenoss.com

1.8. Defects Fixed in this Release

See the document titled "Zenoss Service Dynamics Resolved Defects" (appended to these release notes) for the list of defects that were resolved in this release of Resource Manager.

Zenoss Service Dynamics Version 4.1.1 Fixed Defects

| Defect ID | Summary |
|-----------|---|
| 700 | Error accessing localhost collector without MultiRealm installed |
| 2476 | Errors modeling bonded interfaces on HP-UX v11.31 |
| 2706 | No way to model a vmware infrastructure from UI |
| 27724 | Unicode error if devices have non-latin letters in properties |
| 28633 | HPUX zenpack support for Itanium processors |
| 28686 | Installing Zenoss with remote ZenDS creates zenhub.conf with invalid data |
| 29084 | IE9 event details window cannot resize and scroll bars disappear |
| 29094 | Don't log stacktrace at error level in event log when user enters a bad regex |
| 29112 | Text on 'Reset to Default Settings' button is cut |
| 29118 | Interfaces page: Event rainbow does not automatically refresh with new events |
| 29141 | All monitored components report is always empty |
| 29142 | zenprocess sends erroneous process down events |
| 29168 | Networks side panel gets no scroll bar (screen shot included) |
| 29169 | Component Interface monitoring not working |
| 29176 | Traceback while a user adds a VMware Infrastructure |
| 29183 | Core ZenPack version needs update |
| 29184 | ZenPack version needs update |
| 29186 | Model batch fails when Zenoss has a dynamic service created |
| 29195 | zenhub can't process calls from invalidation workers if hub workers are enabled |
| 29199 | Hub invalidation workers need to be in 4.0.x and 4.1.x |
| 29229 | zenvmwaremodeler - TypeError: not enough arguments for format string |
| 29232 | AMQP message delivery_mode not consistently set to persistent |
| 29233 | zenbatchload needs a corresponding "zenbatchdump" |
| 29234 | Ugly /Cmd/Fail events for NetApp Filers: [Failre instance: Traceback (failure .. |
| 29238 | Config cycle on daemons is never updated |
| 29239 | Memory leak in Python CPP protobuf bindings |
| 29242 | Need a way to log what jobs specific workers are doing |
| 29250 | Graph Range & reset dont work after 3.2.1 update |
| 29270 | zenetl consumes a lot of memory when extracting model batches |
| 29271 | configsipsize parameter disables collection on all but last sipsize devices |
| 29273 | zenping memory leak in pyraw native C code |
| 29274 | Graphs with Width > 500 are not shown completely |
| 29275 | mysqladmin broken for zenoss user "/opt/zends/bin/.mysqladmin: unknown variable 'max_allowed_packet=64M'" |
| 29277 | zenping timeout/retry/maxping preferences have no effect |
| 29278 | zenping invalid network topology leads to spurious ping downs (suppressed) |
| 29279 | zenping sends spurious UP/DOWN events when configs are updated |

| | |
|-------|---|
| 29280 | Configuration Property page slow on initial load |
| 29281 | Reports page automatically expand the report tree recursively |
| 29286 | No way to edit Device and Component Grid Buffer size |
| 29287 | Host Resources MIB reports incorrect values for disks larger than 16TB |
| 29289 | Remove percentage sign from root cause confidence UI |
| 29292 | Component filtering too restrictive in some cases |
| 29296 | Bad community names stop monitoring of devices ever after |
| 29299 | Invalidation workers leak memory |
| 29301 | @deprecated breaks original code |
| 29302 | zenperfsnmp improvements |
| 29303 | Device Utilization 'Show Report' broken |
| 29305 | ZEP event aging and archiving limits are set too conservative |
| 29307 | Infrastructure Page performance |
| 29322 | [unit test] Run Unit Tests on a Zenoss installation with remote ZenDS retrieves several Warnings and Errors |
| 29324 | Invalidation workers still leak |
| 29328 | Error on Zenoss UI after a fresh install |
| 29330 | An error was caught in the zenprocess log after a fresh install |
| 29331 | IOError: zenhubiworker.conf is missing after a fresh install |
| 29332 | Triggers based on multi-valued details not working as expected |
| 29334 | Starting zenping on a machine without IPv6 generates unusual output |
| 29336 | Going back to Infrastructure if filter is applied yields blank grid |
| 29337 | zenvmwaremodeler fails when modeling a NIC with no macAddress |
| 29339 | Nortel Passport interfaces are being assigned the wrong monitoring template |
| 29347 | zenwebtx datasources do not work on components |
| 29352 | Basic Reports buttons are disable for all users |
| 29353 | User "brian", qa-cent5-64-5 now a part of all installations (at least 4.1.1) |
| 29359 | Nexus 7k modeling plugin fails on single VDC |
| 29361 | No events are being forwarded after installing GOM |
| 29362 | Cannot start zenhub with workers |
| 29365 | Call home slows UI |
| 29366 | MySQL max idle connections should be 24h by default (currently 1 year) |
| 29367 | "Unable to create the initial Zenoss object database" 4.1.1 build 1356 broken |
| 29368 | Traceback related to collectors during the Zenoss Installation |
| 29370 | snmp zprop defaults are not ideal |
| 29373 | Error in ZEP API return value when deleting a single configuration value |
| 29374 | zenmodeler crashes when ifconfig's netmask is 0 |
| 29382 | Ack / UnAck events in event console |
| 29388 | /Perf/Filesystem/NetApp transform needs to be more defensive |
| 29389 | Gaps on perf graphs for WMI, AIX, SSH and /Server/Linux device classes |

| | |
|-------|---|
| 29390 | 'Add Template' dialog is unusable with lots of device classes |
| 29396 | Global DNS resolution preference |
| 29398 | Zenoss 4.1, gomd traces back and dies when 'update model' is enabled |
| 29401 | Tracking multiple mysqld's on host fails |
| 29402 | Notifications of impact events do not work |
| 29404 | Warning in Nexus 7k modeling output |
| 29406 | Nexus 7k modeling fails on newer versions of NX-OS (XML namespace changed) |
| 29414 | Cannot save Configuration Property on a device class |
| 29424 | Use 2 hub workers by default |
| 29425 | ip changes not picked up by zenping/global ping |
| 29436 | Random gaps are present on machines graphs |
| 29437 | zenprocess fixes |
| 29440 | zenping not working |
| 29442 | Make sensible grid defaults |
| 29444 | gomd tracebacks when 'update model' is enabled |
| 29445 | Traceback when user tries to add invalid network address |
| 29448 | LDAPAuthenticator increments badPwdCount by 4 on LDAP server |
| 29455 | Session objects limit is too low for a normal install with persistent sessions |
| 29457 | Some vmware host interfaces don't have data |
| 29459 | Zenping datasource should have round-trip-time (rtt), minRtt, maxRtt, standard deviation and packet loss |
| 29468 | Cannot model IPv6 devices |
| 29469 | Zope should use persistent sessions by default |
| 29472 | Devices list is empty |
| 29474 | zenperfsnmp Agent Down event is misleading |
| 29475 | Device Modifications in 4.x have been moved but document reflects old location |
| 29476 | Model batch fails |
| 29477 | zenperfsnmp device events contain an "snmp" component |
| 29480 | Calls to collectors aren't deduped |
| 29482 | Add zInterfaceMapIgnoreDescriptions property to ignore interfaces based on description |
| 29487 | "Worker reports Unhandled Error" traceback when creating a remote collector |
| 29490 | Traceback on zenactiond when RemodelHostsOnMigration is triggered. |
| 29493 | Audit: New things are audited (for QA) |
| 29496 | Zenoss (zenping) reports it's own server as ping down |
| 29507 | Device Detail bar in ZenVMware host fails to load/render properly |
| 29509 | nmap zenping uses APR responses for devices on the same segment |
| 29510 | Commands that return no data still send a clear event indicating success that can't be dropped as useless |
| 29514 | modeler hub service stack traces when ZODB commit lock is used by another process |
| 29515 | Zenping does not work against IPv6 addresses |

| | |
|-------|--|
| 29518 | ZEP Deadlock leads to complete system failure |
| 29519 | Unable to add elements to Impact services KeyError: 'targetUid' |
| 29523 | Multi-valued trigger fix causes all parent organizers to show up in event details |
| 29528 | zentune returns a traceback on Events section KeyError: 'mysqlpassword' |
| 29529 | Ack / UnAck in event console returns to top of grid |
| 29530 | DistributedCollector does not set permissions on nmap correctly |
| 29531 | Several tracebacks related to ZenDS caught in the event log file after the upgrade from 3.2.1 to 4.1.1 |
| 29533 | Tracebacks during the zenoss start process after upgrade from 4.1.0 to 4.1.1 |
| 29534 | GOM Fails To Navigate To Device from certain event hyperlinks |
| 29537 | ZenHub worker throws exception when processing a device entry that's None |
| 29538 | vmware modeler incorrectly marking infrastructure as decommissioned |
| 29539 | Audit: Cannot specify audit level anymore |
| 29540 | Race condition on infrastructure page |
| 29541 | ZEP handling of out of order events sets incorrect first seen time |
| 29544 | ZenHubWorker stats are not hub specific |
| 29545 | [unit test] 4.1.1 unit tests are broken |
| 29546 | Traceback during upgrade from 4.1.0 to 4.1.1 |
| 29550 | Impact version number needs update |
| 29552 | Moving a device from an organizer with command templates to one without still monitors |
| 29553 | Upgrading with a remote database requires re-applying changes to zenoss_*_pre scripts |
| 29554 | Audit: Remodel VM ware endpoint Infraestructure is not being shown in audit logs |
| 29560 | Dropdown too narrow to see graph names when "adding datapoints to graphs" |
| 29563 | Event class actions are not audited |
| 29566 | IE8: Overlapped text on "Set Up Initial Users" |
| 29572 | "zenwebserver deploy -n" not working if n > 1 |
| 29576 | zenwebserver starting fails after the upgrade from 3.2.1 to 4.1.1 |
| 29577 | zenwebserver restart shows down when it's actually up |
| 29578 | extdirect.js does not include enough information for python client library |
| 29580 | Remodel VMware output is not working |
| 29581 | zeneventserver startup is too slow |
| 29582 | When GOM update model is specified, events aren't tagged with device |
| 29584 | UI is broken after the upgrade from 4.1.0 to 4.1.1 |
| 29585 | Nmap --max-retries=0 causes false ping downs |
| 29586 | Wrong HTTP method in ZepConfigClient setConfigValue |
| 29587 | zenwebserver deploy shows down when it's actually up |
| 29589 | zenwebserver reports not listening when the port is used by nginx |

| | |
|-------|---|
| | worker, not master |
| 29594 | Audit: Command-line isn't audited [for QA] |
| 29595 | VM left on wrong host after 2nd vmotion |
| 29596 | vmotion does not result in remodeling if collector name differs from its hostname |
| 29597 | Modeling a vmware host with no vms causes zenvmwaremodeler to fail |
| 29604 | Zentune: Global settings are not being set with default values |
| 29607 | Collector calls can overload the hub, so rabbit fills up /var |
| 29608 | ERROR Application Couldn't install Products.ZenUI3 |
| 29610 | zenwebserver shows down on the Zenpacks page |
| 29617 | Cloned vm does not get add to VMware infrastructure |
| 29619 | Adding a guest or datastore doesn't register until a full remodel |
| 29624 | 4.1.1 three zenpacks need version bumps |
| 29625 | Traceback during the installation of 4.1.1 related to "zen.ZenPackCmd" |
| 29627 | zentune doesn't know about zenwebserver |
| 29635 | ERROR Application Couldn't install Products.ZenUI3 |
| 29640 | VMware infrastructure doesn't get perf data on remote collector |
| 29641 | Email action in zenactiond should send emails in batches |
| 29642 | IPInterface Component grid breaks when RRD file not available |
| 29649 | PIP in the Interface Utilization Report under Enterprise Reports |
| 29652 | Zenping not working on local collectors after upgrade |
| 29653 | Maintenance windows stop working if an exception occurs |
| 29654 | OSProcess regex monitoring is broken |
| 29656 | Infrastructure: Unable to see devices at the end of the scroll list |
| 29660 | Getting traceback while running zentune on an ISO appliance installation |
| 29662 | UI in the Criteria tab of GOM is broken in Fire Fox |
| 29676 | Component filter does not work at all |

Chapter 1. zenwebserver

1.1. About

Use zenwebserver to deploy and manage multiple Zope instances. It includes a software load balancer (nginx), and replaces zopectl for Zenoss Service Dynamics users.

1.2. Prerequisites

| Prerequisite | Restriction |
|-------------------|--------------------------|
| Product | Resource Manager 4.1.1 |
| Required ZenPacks | ZenPacks.zenoss.WebScale |

Table 1.1. Prerequisites

1.3. Installation

To use zenwebserver, install the WebScale ZenPack.

Note

This ZenPack is included by default in the Zenoss Enterprise ZenPacks RPM.

Enter this command:

```
zenpack --install ZenPacks.zenoss.WebScale-1.0.0-py2.7-Platform-Architecture.egg
```

The installation process replaces zopectl in the startup script (or in the `daemons.txt` file) with zenwebserver. After installation, use zenwebserver as the control script to manage the application server.

Note

If you have multiple Zope instances deployed behind a custom load balancer setup, installation of this ZenPack will not install zenwebserver as your UI control script. You must install it manually after determining and executing your migration strategy.

1.4. Usage

zenwebserver *Arguments Options Targets*

1.4.1. Arguments

Valid arguments are:

- **run** - Starts Zope in the foreground, on the port normally used by the load balancer. Neither the load balancer nor other Zope servers are used.
- **start** - Starts the load balancer and Zope servers. If any are running already, they are ignored.
- **stop** - Stops the load balancer and Zope servers. If any are stopped already, they are ignored.
- **restart** - Stops and then restarts the load balancer and Zope servers. To minimize downtime, the load balancer is restarted first, and then each Zope server in turn. This ensures that the Zope server pool is never empty.
- **status** - Provides status information. It prints the status of the load balancer, including its PID.
- **deploy** - Creates or destroys Zope instances. It adds or removes instances from the server pool and updates the load balancer to reference the altered server pool. If the load balancer is running already, then its configuration is reloaded without stopping it.

- **reload** - Reloads the load balancer configuration. For example, if you make a change to the nginx configuration to listen at a different port, reload it to use the new port without restarting.
- **attach** - Returns a detached Zope server to the server pool and updates the load balancer.
- **detach** - Removes a Zope server from the server pool and updates the load balancer. (Zope continues to run, but does not get traffic from the load balancer.)
- **debug** - Deploys a Zope server without adding it to the server pool, starting it immediately in the foreground. This server can only be accessed directly. The server is automatically destroyed upon exiting the process.
- **help** - Returns command usage information.

1.4.2. Options

Valid options are:

- **-v** - Prints more information, including the status of each Zope server, the ports at which the processes are listening, and the servers currently detached from the server pool.

1.4.3. Targets

Several commands accept one or more targets against which the command should be executed. If you do not specify a target, the command runs the action against all targets.

Valid targets are:

- **loadbalancer** - Load balancer. Alternatively, you can specify:

```
nginx
```

- **servers** - All Zope servers.
- **server n** - Specific Zope server, where n is the server number. Alternatively, you can specify just a server number or numbers. For example, both of the following commands stop Zope servers 2 and 3:

```
zenwebserver stop server2 server3
```

```
zenwebserver stop 2 3
```

1.4.4. Command Use and Examples

Status

```
zenwebserver status [-v]
```

Start, Stop, and Restart

```
zenwebserver {stop|start|restart} [-v] [Targets]
```

Manage the Number of Zope Servers

```
zenwebserver deploy {n|-n|+n}
```

Examples:

- `zenwebserver deploy 5` # Ensures that exactly 5 Zope servers are running.
- `zenwebserver deploy +1` # Deploys one additional Zope server, regardless of the current number.
- `zenwebserver deploy -3` # Destroys up to 3 Zope servers (as long as the minimum of 1 is maintained).

Manage the Server Pool

```
zenwebserver {attach|detach} Targets
```

Detaching a target is useful when you want to isolate a Zope server and access it via its direct port to ensure that your requests are the only ones being handled by that server.

Start an Independent Instance

zenwebserver debug