Zenoss®

Transforming IT Operations

Zenoss Service Dynamics Extended Monitoring

Release 4.2.5

Zenoss, Inc.

www.zenoss.com

Zenoss Service Dynamics Extended Monitoring

Part Number: 27-032014-4.2-v13

Zenoss, Inc.
11305 Four Points Drive
Bldg 1 - Suite 300
Austin, Texas 78726

About this guide

This guide describes the ZenPacks that are distributed with, or available for, Zenoss Service Dynamics Resource Manager (Resource Manager).

This release includes new ZenPacks that deprecate previous ZenPacks. Information about both the new and deprecated ZenPacks are included, to facilitate transition to the new ZenPacks.

Unlike previous editions of this guide, chapters are ordered by ZenPack name.

Two new appendices are included.

Administrating ZenPacks **on page 252**

This appendix includes procedures for installing, upgrading, and removing ZenPacks.

Resource Manager daemons **on page 256**

This appendix includes descriptions of the daemons that are always present in Resource Manager, and the daemons that are installed when specific ZenPacks are installed.

ZenPack Index

The following table lists the ZenPacks that are included in this release of Resource Manager, and some that are available on the *Zenoss Support* site. The table includes

- The ZenPack name and a link to its documentation, if any
- A brief description of the ZenPack
- A one-character code indicating the ZenPack type

P

Platform ZenPacks add features to Resource Manager or enhance existing features.

M

Monitoring ZenPacks add device modeling and data collection capabilities for specific device types.

Some platform ZenPacks are undocumented because they provide basic, background services.

Name and location	Description	Type
(ActiveDirectory) Active Directory on page 9	The ZenPacks.zenoss.ActiveDirectory ZenPack uses WMI to monitor Microsoft Active Directory authentication metrics. NOTE: This ZenPack is deprecated; see *(Microsoft.Windows) Microsoft Windows* on page 123.	M
(AdvancedSearch) Advanced Search on page 11	The ZenPacks.zenoss.AdvancedSearch ZenPack enables locating devices and other system objects, as well as events and services.	P
(AixMonitor) AIX on page 13	The ZenPacks.zenoss.AixMonitor ZenPack uses Secure Shell (SSH) to monitor AIX hosts.	M
(AuditLog)	The Zenpacks.zenoss.AuditLog ZenPack is undocumented.	P
(AutoTune) ZenTune on page 22	The ZenPacks.zenoss.AutoTune ZenPack provides the ZenTune "tuning advisor" feature of Resource Manager.	P
(BigIpMonitor) BIG-IP Devices on page 32	The ZenPacks.zenoss.BigIpMonitor ZenPack monitors BIG-IP devices (from F5 Networks).	M
(BrocadeMonitor) Brocade SAN Switches on page 34	The ZenPacks.zenoss.BrocadeMonitor ZenPack monitors Brocade Storage Area Network (SAN) switches.	M

Name and location	Description	Type
(CatalogService)	The Zenpacks.zenoss.CatalogService ZenPack is undocumented.	P
(CheckPointMonitor) Check Point Security Appliance on page 36	The ZenPacks.zenoss.CheckPointMonitor ZenPack monitors security appliances from Check Point.	M
(CiscoMonitor) Cisco Devices on page 38	The ZenPacks.zenoss.CiscoMonitor ZenPack monitors faults and performance of a wide range of Cisco equipment, including virtual resources such as virtual firewalls and virtual load balancers.	M
(CiscoUCS) Cisco UCS on page 46	The ZenPacks.zenoss.CiscoUCS ZenPack uses HTTP to monitor Cisco Unified Computing System (UCS) devices.	M
Oracle Database	The ZenPacks.zenoss.DatabaseMonitor ZenPack monitors Oracle database servers. NOTE: This ZenPack is not installed when Resource Manager is installed. To download it, visit the *Zenoss Support* site.	M
(DellMonitor) Dell Hardware on page 53	The ZenPacks.zenoss.DellMonitor ZenPack provides customized modeling of devices running Dell OpenManage agents, and includes identification for proprietary Dell hardware.	M
(DeviceSearch)	The Zenpacks.zenoss.DeviceSearch ZenPack is undocumented.	P
(Diagram) Datacenter View on page 55	The ZenPacks.zenoss.Diagram ZenPack enables a visual representation of devices (such as servers or blades) and device containers (such as racks or chassis).	P
(DigMonitor) Dig Monitor on page 60	The ZenPacks.zenoss.DigMonitor ZenPack monitors the response time of DNS lookups.	M
(DistributedCollector) Distributed Collector on page 62	The ZenPacks.zenoss.DistributedCollector ZenPack allows you to deploy additional performance collection and event monitoring daemons, on the Resource Manager master host, and on other hosts.	P
(DnsMonitor) DNS Monitor on page 69	The ZenPacks.zenoss.DnsMonitor ZenPack monitors the response time of DNS requests.	M
(DynamicView) Dynamic Service View on page 71	The ZenPacks.zenoss.DynamicView ZenPack provides a dynamic visualization of system objects and their relationships to other objects.	P
(EnterpriseCollector) Enterprise Collector on page 75	The ZenPacks.zenoss.EnterpriseCollector ZenPack allows several collector daemons to start and to monitor devices, even if a connection to `zenhub` is not available when a collector daemon starts.	P
(EnterpriseLinux) Enterprise Linux on page 77	The ZenPacks.zenoss.EnterpriseLinux ZenPack extends the capabilities of ZenPacks.zenoss.LinuxMonitor and enables Resource Manager to use Secure Shell (SSH) to monitor Linux hosts.	M
(EnterpriseReports) Enterprise Reports on page 81	The ZenPacks.zenoss.EnterpriseReports ZenPack adds a variety of reports to Resource Manager.	P
(EnterpriseSecurity) Enterprise Security on page 84	The ZenPacks.zenoss.EnterpriseSecurity ZenPack enhances Resource Manager security by encrypting stored passwords.	P
(EnterpriseSkin)	The Zenpacks.zenoss.EnterpriseSkin ZenPack is undocumented.	P

Name and location	Description	Type
(EsxTop) VMware ESX Server on page 85	The ZenPacks.zenoss.EsxTop ZenPack uses the VMware `resxtop` command to gather performance information about VMware Infrastructure™ ESX™ servers. NOTE: This ZenPack is deprecated; see *(vSphere) VMware vSphere* on page 177.	M
(FoundryMonitor) Foundry Networks Devices on page 88	The ZenPacks.zenoss.FoundryMonitor ZenPack monitors networking devices built by Foundry Networks (now Brocade Communication Systems). NOTE: This ZenPack is not installed when Resource Manager is installed. To download it, visit the *Zenoss Support* site.	M
(FtpMonitor) FTP Monitor on page 90	The ZenPacks.zenoss.FtpMonitor ZenPack monitors the response times of File Transfer Protocol (FTP) server connection requests.	M
(HPMonitor) HP Monitor on page 93	The ZenPacks.zenoss.HPMonitor ZenPack provides customized modeling of devices running HP Insight Management Agents, and includes identification for proprietary HP hardware.	M
(HpuxMonitor) HP-UX Monitor on page 95	The ZenPacks.zenoss.HpuxMonitor ZenPack monitors HP UNIX (HP-UX) servers through Secure Shell (SSH).	M
(HttpMonitor) HTTP Monitor on page 99	The ZenPacks.zenoss.HttpMonitor ZenPack monitors the response times of HTTP server connection requests, and determines whether specific content exists on a Web page.	M
(IISMonitor) Microsoft Internet Information Server on page 102	The ZenPacks.zenoss.IISMonitor ZenPack uses Windows Perfmon to collect performance metrics from Microsoft Internet Information Server (IIS). NOTE: This ZenPack is deprecated; see *(Microsoft.Windows) Microsoft Windows* on page 123.	M
(IRCDMonitor) IRCD Monitor on page 104	The ZenPacks.zenoss.IRCDMonitor ZenPack monitors the number of users connected to an Internet Relay Chat (IRC) server.	M
(JabberMonitor) Jabber Monitor on page 106	The ZenPacks.zenoss.JabberMonitor ZenPack monitors the response times of Jabber instant messaging servers.	M
(JBossMonitor) JBoss Application Server on page 108	The ZenPacks.zenoss.JBossMonitor ZenPack monitors JBoss application servers.	M
(JuniperMonitor) Juniper Monitor on page 112	The ZenPacks.zenoss.JuniperMonitor ZenPack monitors devices from Juniper Networks.	M
(LDAPAuthenticator) LDAP Authentication on page 114	The ZenPacks.zenoss.LDAPAuthenticator ZenPack enables pass-through authentication to external LDAP-based servers such as Microsoft Active Directory or OpenLDAP.	P
(LDAPMonitor) LDAP Monitor on page 119	The ZenPacks.zenoss.LDAPMonitor ZenPack monitors the response time of Lightweight Directory Access Protocol (LDAP) servers.	M
(Licensing)	The Zenpacks.zenoss.Licensing ZenPack is undocumented.	P
(LinuxMonitor) Linux Monitor on page 121	The ZenPacks.zenoss.LinuxMonitor ZenPack demonstrates how to develop new plugins that collect performance data using Secure Shell.	P
(Microsoft.Windows) Microsoft Windows on page 123	The ZenPacks.zenoss.Microsoft.Windows ZenPack monitors Microsoft Windows systems and services through the Windows	M

Name and location	Description	Type
	Remote Management (WinRM) and Windows Remote Shell (WinRS) interfaces.	
(MSExchange) Microsoft Exchange on page 136	The ZenPacks.zenoss.MSExchange ZenPack uses WMI to monitor Microsoft Exchange and related services. NOTE: This ZenPack is deprecated; see *(Microsoft.Windows) Microsoft Windows* on page 123.	M
(MSMQMonitor) Microsoft Message Queueing on page 138	The ZenPacks.zenoss.MSMQMonitor ZenPack uses WMI to automatically discover Microsoft Message Queuing (MSMQ) queues, and monitor the number of messages queued in each. NOTE: This ZenPack is deprecated; see *(Microsoft.Windows) Microsoft Windows* on page 123.	M
(MSSQLServer) Microsoft SQL Server on page 140	The ZenPacks.zenoss.MSSQLServer ZenPack uses WMI to monitor Microsoft SQL Server and its related services. NOTE: This ZenPack is deprecated; see *(Microsoft.Windows) Microsoft Windows* on page 123.	M
(MultiRealmIP) Multi-Realm IP Networks on page 142	The ZenPacks.zenoss.MultiRealmIP ZenPack extends the modeling, monitoring, and event management features of Resource Manager to accommodate overlapping IP namespaces. NOTE: This ZenPack is not installed when Resource Manager is installed. To download it, visit the *Zenoss Support* site.	P
(MySqlMonitor) MySQL Database Monitor on page 145	The ZenPacks.zenoss.MySqlMonitor ZenPack monitors MySQL database servers through the Python `twisted.enterprise.adbapi` asynchronous framework.	M
NetApp Monitor	The ZenPacks.zenoss.NetAppMonitor ZenPack monitors devices from NetApp.	M
(NetScreenMonitor) NetScreen Monitor on page 150	The ZenPacks.zenoss.NetScreenMonitor ZenPack monitors devices from NetScreen Technologies.	M
(NNTPMonitor) NNTP Monitor on page 152	The ZenPacks.zenoss.NNTPMonitor ZenPack monitors the response time of Network News Transfer Protocol (NNTP) servers.	M
(NtpMonitor) NTP Monitor on page 156	The ZenPacks.zenoss.NtpMonitor ZenPack monitors the difference between the system time a server is using and the time a Network Time Protocol (NTP) server is reporting.	M
Zenpacks.zenoss.PythonCollector	This ZenPack provides a Python data source type and polling daemon (**zenpython**) for use in customized ZenPacks.	P
(RANCIDIntegrator) RANCID Integration on page 157	The ZenPacks.zenoss.RANCIDIntegrator ZenPack enables integration between the RANCID configuration management tool and Resource Manager.	P
RPCMonitor (RPC Monitor) on page 159	The ZenPacks.zenoss.RPCMonitor ZenPack monitors applications that use Open Network Computing (ONC) Remote Procedure Call (RPC). NOTE: This ZenPack is not installed when Resource Manager is installed. To download it, visit the *Zenoss Support* site.	M
Solaris	The ZenPacks.zenoss.SolarisMonitor ZenPack monitors Solaris servers through Secure Shell (SSH) or SNMP.	M

Name and location	Description	Type
Splunk (Splunk) on page 161	The ZenPacks.zenoss.Splunk ZenPack facilitates unified monitoring by invoking customized searches of Splunk databases. NOTE: This ZenPack is not installed when Resource Manager is installed. To download it, visit the *Zenoss Support* site.	M
(StorageBase) Storage Base on page 166	The ZenPacks.zenoss.StorageBase ZenPack contains base classes and reports for ZenPacks that use the base classes.	P
(SugarCRMMonitor) SugarCRM Monitor on page 167	The ZenPacks.zenoss.SugarCRMMonitor ZenPack monitors SugarCRM services.	M
(TomcatMonitor) Apache Tomcat on page 169	The ZenPacks.zenoss.TomcatMonitor ZenPack monitors Apache Tomcat servers.	M
(vCloud) VMware vCloud on page 173	The ZenPacks.zenoss.vCloud ZenPack monitors virtual infrastructure services that are managed by VMware vCloud Suite platforms.	M
(VMwareESXMonitor) VMware ESX SNMP on page 175	The ZenPacks.zenoss.VMwareESXMonitor ZenPack monitors VMware ESX hosts and their guests, using SNMP. NOTE: This ZenPack is deprecated; see *(vSphere) VMware vSphere* on page 177.	M
(vSphere) VMware vSphere on page 177	The ZenPacks.zenoss.vSphere ZenPack monitors VMware vSphere systems and services through a vCenter server, using the vSphere API.	M
ZenPacks.zenoss.WBEM	The ZenPacks.zenoss.WBEM ZenPack provides a WBEM data source type and a WBEMPlugin base modeler plugin, and is used to create customized ZenPacks. This ZenPack is included in the Resource Manager RPM package, and copied to the `$ZENHOME/packs` directory, but not installed.	P
(WebLogicMonitor) WebLogic Monitor on page 184	The ZenPacks.zenoss.WebLogicMonitor ZenPack monitors Oracle WebLogic Server services.	M
(WebScale) WebScale on page 189	The ZenPacks.zenoss.WebScale ZenPack adds the `zenwebserver` daemon, to deploy and manage multiple Zope instances.	P
(WebsphereMonitor) IBM WebSphere on page 193	The ZenPacks.zenoss.WebsphereMonitor ZenPack monitors IBM WebSphere Application Servers (WAS).	M
WindowsMonitor (Microsoft Windows) on page 197	The ZenPacks.zenoss.WindowsMonitor ZenPack uses WMI to monitor the performance of Microsoft Windows servers. NOTE: This ZenPack is deprecated; see *(Microsoft.Windows) Microsoft Windows* on page 123.	M
(XenMonitor) Xen Virtual Hosts on page 205	The ZenPacks.zenoss.XenMonitor ZenPack monitors Xen para-virtualized domains.	M
(ZenDeviceACL) Device ACLs on page 207	The ZenPacks.zenoss.ZenDeviceACL ZenPack adds fine-grained device access controls (ACLs) to Resource Manager.	P
(ZenHoltWinters) Predictive Thresholding on page 210	The ZenPacks.zenoss.ZenHoltWinters ZenPack adds the ability to create threshold events when a device exceeds cyclical predicted values.	P

Name and location	Description	Type
(ZenJMX) Java Management Extensions on page 212	The ZenPacks.zenoss.ZenJMX ZenPack adds the `zenjmx` daemon, which communicates with remote Java Management Extensions (JMX) agents, to collect data from Java-based applications.	P
(ZenMailTx) Mail Transactions on page 225	The ZenPacks.zenoss.ZenMailTx ZenPack allows you to monitor round-trip email delivery.	M
(ZenOperatorRole) Operator Role on page 227	The ZenPacks.zenoss.ZenOperatorRole ZenPack adds the `ZenOperator` user role.	P
(ZenSQLTx) SQL Transactions on page 228	The ZenPacks.zenoss.ZenSQLTx ZenPack monitors the availability and performance of MySQL, Sybase and Microsoft SQL servers.	M
(ZenossVirtualHostMonitor)	The Zenpacks.zenoss.ZenossVirtualHostMonitor ZenPack is undocumented.	P
(ZenVMware) VMware vSphere on page 233	The ZenPacks.zenoss.ZenVMware ZenPack monitors VMware devices through the vSphere API. NOTE: This ZenPack is deprecated; see *(vSphere) VMware vSphere* on page 177.	M
(ZenWebTx) Web-Based Synthetic Transactions on page 241	The ZenPacks.zenoss.ZenWebTx ZenPack adds the `zenwebtx` daemon, which enables availability and performance monitoring of web sites through synthetic HTTP transactions.	M

1 (ActiveDirectory) Active Directory

The ZenPacks.zenoss.ActiveDirectory ZenPack uses WMI to monitor Microsoft Active Directory authentication metrics.

Note This ZenPack is deprecated; see *(Microsoft.Windows) Microsoft Windows* on page 123.

This ZenPack creates a device class for Microsoft Active Directory with appropriate priorities. It also creates a Windows Service class and IP Service class for Active Directory-related services with monitoring enabled.

Use this ZenPack to monitor the following metrics:

- DS Client Binds/Sec
- DS Directory Reads/Sec, Searches/Sec and Writes/Sec
- DS Monitor List Size
- DS Name Cache Hit Rate
- DS Notify Queue Size
- DS Search Sub-operations/Sec
- DS Server Binds/Sec, Server Name Translations/Sec
- DS Threads In Use
- KDC AS Requests, TGS Requests
- Kerberos Authentications
- LDAP Active Threads
- LDAP Bind Time
- LDAP Client Sessions
- LDAP New / New SSL and Closed Connections/Sec
- LDAP Searches/Sec, Writes/Sec
- LDAP Successful Binds
- LDAP UDP Operations/Sec
- NTLM Authentications

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x
Required ZenPacks	ZenPacks.zenoss.WindowsMonitor,

Prerequisite	Restriction
	ZenPacks.zenoss.ActiveDirectory

Enable Monitoring

All Active Directory services must have a device entry under the `/Devices/Server/Windows/Active Directory` device class. In addition, verify that your Resource Manager Windows service account has access to the Active Directory service.

1 Navigate to the device or device class in the Resource Manager interface.

- If applying changes to a device class:
  1 Select the class in the devices hierarchy.
  2 Click **Details**.
  3 Select Configuration Properties.
- If applying changes to a device:
  1 Click the device in the device list.
  2 Select Configuration Properties.

2 Verify the credentials for the service account to access the service.

Table 1: Active Directory Configuration Properties

Name	Description
zWinUser	Windows user with privileges to gather performance information.
zWinPassword	Password for the above user.

3 Click **Save** to save your changes.

You will now be able to start collecting the Active Directory server metrics from this device.

4 Navigate to Graphs and you should see some placeholders for graphs. After approximately fifteen minutes you should see the graphs start to become populated with information.

Note For more information about user credentials and troubleshooting WMI connections, see *WindowsMonitor (Microsoft Windows)* on page 197.

Daemons

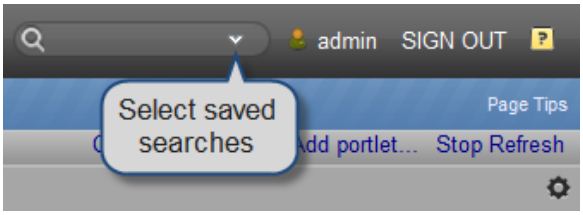
Type	Name
Performance Collector	`zenwinperf`

(AdvancedSearch) Advanced Search 2

The ZenPacks.zenoss.AdvancedSearch ZenPack enables locating devices and other system objects, as well as events and services.

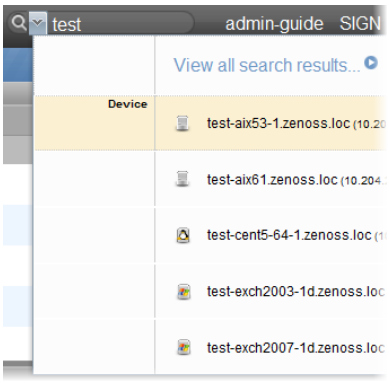
In the user interface, the advanced search feature is located adjacent to the user information area.

Figure 1: Advanced Search (User Information Area)



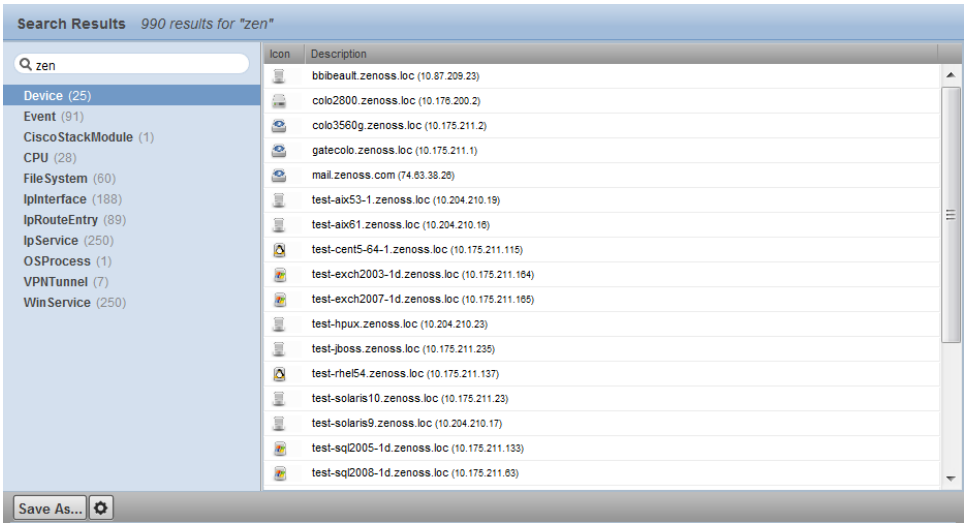
To search, enter part or all of a name in the search box. The system displays matches, categorized by type.

Figure 2: Search Results



To view all search results, click the indicator at the top of the list. The full list of results appears.

Figure 3: All Search Results



From here, you can display search results by category. Click in the left panel to filter search results by a selection.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x
Required ZenPacks	ZenPacks.zenoss.AdvancedSearch

Working with Saved Searches

To save a search:

1 Click **Save As**.

the Save Search As dialog appears.

2 Enter a name for the saved search, and then click **Submit**.

To retrieve a saved search, select it from the search box menu.

You also can manage saved searches. Access all saved searches from two locations:

- Search box menu
- Action menu located at the bottom of the Search Results page

The Manage Saved Searches dialog lets you view the queries associated with saved searches and delete saved searches.

(AixMonitor) AIX 3

The ZenPacks.zenoss.AixMonitor ZenPack uses Secure Shell (SSH) to monitor AIX hosts.

This ZenPack models and monitors devices placed in the `/Server/SSH/AIX` device class by running commands and parsing the output. The account used to monitor the device does not require root access or special privileges.

This ZenPack provides:

- File system and process monitoring
- Network interfaces and route modeling
- CPU utilization information
- Hardware information (memory, number of CPUs, machine serial numbers, model numbers)
- OS information (OS level command style information)
- LPP and RPM information (such as installed software)

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x
Required ZenPacks	ZenPacks.zenoss.AixMonitor
AIX Releases Supported	5.3 and 6.1

Note If using a distributed collector setup, SSH requires firewall access (default of port 22) from the collector to the monitored server.

Add an AIX Server

The following procedure assumes that the credentials have been set.

1. From Infrastructure > Devices, select Add a Single Device.
2. Enter the following information in the dialog:

Table 2: Adding AIX Device Information

Name	Description
Name or IP	AIX host to model
Device Class	`/Server/SSH/AIX`
Model Device	Select this option unless adding a device with username/password different than found in the device class. If you do not select this option, then you must add the credentials, and then manually model the device.

3 Click **Add Device** to add the device.

Related Links

Set AIX Server Monitoring Credentials on page 14

Set AIX Server Monitoring Credentials

All AIX servers must have a device entry in an organizer below the `/Devices/Server/SSH/AIX` device class.

Note The SSH monitoring feature will attempt to use key-based authentication before using a configuration properties password value.

1 Navigate to the device class or device.

- If applying changes to a device class:
  1 Select the class in the devices hierarchy.
  2 Click **Details**.
  3 Select Configuration Properties.
- If applying changes to a device:
  1 Click the device in the device list.
  2 Select Configuration Properties.

2 Verify the credentials for the service account to access the service.

Table 3: AIX Configuration Properties

Name	Description
zCommandUsername	AIX user with privileges to gather performance information
zCommandPassword	Password for the AIX user

3 Click **Save** to save your changes.

Resolving CHANNEL_OPEN_FAILURE Issues

The zencommand daemon's log file ($ZENHOME/*collector*/zencommand.log) may show messages stating:

```
ERROR zen.SshClient CHANNEL_OPEN_FAILURE: Authentication failure
WARNING:zen.SshClient:Open of command failed (error code 1): open failed
```

If the sshd daemon's log file on the remote device is examined, it may report that the MAX_SESSIONS number of connections has been exceeded and that it is denying the connection request. At least in the OpenSSH daemons, this MAX_SESSIONS number is a compile-time option and cannot be reset in a configuration file.

To work around this limitation of the sshd daemon, use the configuration property *zSshConcurrentSessions* to control the number of connections created by zencommand to the remote device.

1 Navigate to the device or device class in the Resource Manager interface.
   - If applying changes to a device class:
     1 Select the class in the devices hierarchy.
     2 Click **Details**.
     3 Select Configuration Properties.
   - If applying changes to a device:
     1 Click the device in the device list.
     2 Select Configuration Properties.
2 Apply an appropriate value for the maximum number of sessions.

Table 4: Concurrent SSH Configuration Properties

Name	Description
zSshConcurrentSessions	Maximum number of sessions supported by the remote device's MAX_SESSIONS parameter. Common values for AIX are 2 or 10.

3 Click **Save** to save your changes.

Resolving Timeout Issues

The zencommand daemon's log file ($ZENHOME/*collector*/zencommand.log) may show messages stating:

```
WARNING:zen.zencommand:Command timed out on device device_name: command
```

If this occurs, it usually indicates that the remote device has taken too long to return results from the commands. To increase the amount of time for devices to return results, change the configuration property *zCommandCommandTimeout* to a larger value.

1 Navigate to the device or device class in the Resource Manager interface.
   - If applying changes to a device class:
     1 Select the class in the devices hierarchy.
     2 Click **Details**.

3. Select Configuration Properties.

- If applying changes to a device:

1. Click the device in the device list.
2. Select Configuration Properties.

2. Apply an appropriate value for the command timeout.

Table 5: SSH Timeout Configuration Properties

Name	Description
zCommandCommandTimeout	The number of seconds to wait for commands to complete on the remote device.

3. Click **Save** to save your changes.

Daemons

Type	Name
Modeler	`zenmodeler`
Performance Collector	`zencommand`

4 (ApacheMonitor) Apache HTTP Server

The ZenPacks.zenoss.ApacheMonitor ZenPack monitors Apache HTTP Server by collecting metrics through the `mod_status` module.

The following metrics are collected and graphed for Apache HTTP Server.

- Requests per Second
- Throughput (Bytes/sec and Bytes/request)
- CPU Utilization of the HTTP server and all worker processes or threads
- Slot Usage (Open, Waiting, Reading Request, Sending Reply, Keep-Alive DNS Lookup, and Logging)

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.ApacheMonitor

Display the Status Page in Apache Version 1.3 or Higher

1. On the Apache server, locate the `httpd.conf` file. Generally, this file is located at `/etc/httpd/httpd.conf` or `/etc/httpd/conf/httpd.conf`; however, other locations are possible depending on your operating system and setup.

   If you cannot locate the configuration file, use your system's search facilities to locate it. For Windows, use the **Search** button of the Windows Explorer tool. For Unix, try the following command:

   ```
   find / -name httpd.conf
   ```

2. Check to see that the following line is not commented out and is available in `httpd.conf` or `/etc/apache/modules.conf`:

   ```
   LoadModule status_module /usr/lib/apache/1.3/mod_status.so
   ```

   Note You may have to search in alternate locations to find the `mod_status.so` file. Also, the syntax may differ depending on your configuration.

3 Turn the ExtendedStatus option on in the `httpd.conf file`. This option is typically commented out. You can enable it by uncommenting it or ensuring that it is defined.

```
#ExtendedStatus on
```

becomes:

```
ExtendedStatus on
```

4 Enable the `/server-status` location in the `httpd.conf` file. Typically, this option exists but is commented out.

```
#<Location /server-status>
#    SetHandler server-status
#    Order deny,allow
#    Deny from all
#    Allow from .example.com
#</Location>
```

becomes:

```
<Location /server-status>
SetHandler server-status
Order deny,allow
Deny from all
Allow from zenoss.example.com
</Location>
```

Note Your Resource Manager server or servers must be able to connect to your Apache server. Ensure that it is listed here or is part of the network specified in this chunk of configuration.

To specify multiple servers, separate the entries with spaces. If you specify an IP address range rather than a destination, be sure to add a network mask to the end of the IP address range.

The following example allows a server called externalzenoss.example.com, as well as all servers that start with 192.168.10, in their addresses:

```
<Location /server-status>SetHandler server-status
Order deny,allow
Deny from all
Allow from externalzenoss.example.com 192.168.10.0/24
</Location>
```

---

5 Save the `httpd.conf` file with these changes and verify that the configuration file is correct. This can be accomplished with following command.

```
apachectl -t
```

Correct any issues before restarting Apache.

6 Restart the Web server (`httpd`). This can be accomplished with following command.

```
apachectl restart
```

Display the Status Page in Apache Version 2.x

1 On the Apache server, find the `httpd.conf` file. This is usually `/etc/apache2/apache2.conf` or `/etc/apache2/conf/httpd.conf`; however, other locations are possible depending on your operating system and setup.

If you are unsure about where your configuration file is located, use your system;s search facilities to locate this file. Under Windows, use the **Search** button of the Windows Explorer tool. Under Unix, try the following command:

```
find / -name httpd.conf
```

2 Verify that the `mod_status` module is loaded.

```
 apache% apachec2ctl -M 2<&1 | grep status
status_module (shared)
```

The previous output indicates that the module is loaded and no further configuration is necessary. If there is no output, then copy the `mods-available/status.load` to the `mods-enabled` directory, and then run:

```
 apache% /etc/init.d/apache2 force-reload
```

3 Turn the ExtendedStatus option on in the `httpd.conf file`. This option is typically commented out. You can enable it by uncommenting it or ensuring that it is defined.

```
#ExtendedStatus on
```

becomes:

```
ExtendedStatus on
```

4 Enable the `/server-status` location in the `httpd.conf` file. This is another option that typically already exists but is commented out.

```
#<Location /server-status>
#    SetHandler server-status
#    Order deny,allow
#    Deny from all
#    Allow from .example.com
#</Location>
```

becomes:

```
<Location /server-status>
SetHandler server-status
Order deny,allow
Deny from all
Allow from zenoss.example.com
</Location>
```

Note Your Resource Manager server or servers must be able to connect to your Apache server so you must ensure that it is either listed here or is a part of the network specified in this chunk of configuration.

To specify multiple servers, separate the entries with spaces. If you would like to specify an IP address range rather than a destination, be sure to add a network mask to the end of the IP address range. The following

example allows a server called externalzenoss.example.com as well as all servers that start with '192.168.10' in their addresses:

```
<Location /server-status>SetHandler server-status
Order deny,allowDeny from all
Allow from externalzenoss.example.com 192.168.10.0/24
</Location>
```

5 Save the `httpd.conf` file with these changes and verify that the configuration file is correct. This can be accomplished with following command.

```
apache2ctl -t
```

Correct any issues before restarting Apache.

6 Restart the webserver (`httpd`). This can be accomplished with following command.

```
apache2ctl restart
```

Verifying Your Apache Configuration

Once Apache has been configured, you should verify that it is working correctly. To verify your Apache server, point your Web browser to your Apache server at the appropriately modified URL:

`http://`*your-apache-server*`/server-status?auto`

This is an example of what you might see:

```
Total Accesses: 1
Total kBytes: 2
Uptime: 43
ReqPerSec: .0232558
BytesPerSec: 47.6279
BytesPerReq: 2048
BusyWorkers: 1
IdleWorkers: 5
Scoreboard: _W____.............................
```

If there is a configuration issue, you should see an error message telling you that the page is forbidden.

Note Your Resource Manager server or servers must be able to connect to your Apache server by using HTTP to receive information. This means that the Resource Manager server must be permitted not only by the Apache configuration settings, but also by any firewalls or proxies between the Resource Manager server and the Apache server, including any firewall on the Apache server. If there are any proxies, they must be configured to allow the Resource Manager HTTP traffic through. Consult your network administrator and security officer to verify the firewall configuration and your site's policies.

Further note that the name or IP address that your server has behind a firewall may be different than the IP address (some form of Network Address Translation (NAT)) or name resolution (the way that the external server resolves names may be through an Internet-visible DNS system rather than an intranet-only DNS system).

Configure Resource Manager to Monitor the Web Server

Once the Apache server is configured to allow Resource Manager to access the extended status, you can add Apache monitoring to the device within Resource Manager by binding the Apache template to the device.

1. Select Infrastructure from the navigation bar.
2. Click the device name in the device list.

   The device overview page appears.
3. In the left panel, expand Monitoring Templates, and then select Device.
4. Select Bind Templates from the Action menu.

   The Bind Templates dialog appears.
5. Add the Apache template to the list of templates, and then click **Save**.

   The Apache template is added. The system can now begin collecting the Apache server metrics from this device.

Daemons

Type	Name
Performance Collector	`zencommand`

(AutoTune) ZenTune 5

This ZenPack provides the ZenTune "tuning advisor" feature of Resource Manager.

This ZenPack analyzes your system configuration and makes recommendations for better performance.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.2
Required ZenPacks	ZenPacks.zenoss.AutoTune

Configuring ZenTune

You can set values for several options in the `zentune.conf` configuration file (or when running ZenTune from the command line) to configure behavior.

When setting up ZenTune, you can define options to send a test event through the lifecycle to make sure that Zenoss is processing events before the timeout. If it fails to process in time, an email can be sent out.

- `testevent-enable` -- When testevent-enable is present in the `zentune.conf` file, a test event will be sent. If it is not present or commented out, no test event is sent.
- `testevent-email` *ValidEmailAddress* -- If testevent-enable is present and the test event times out, an email will be sent to the defined email address.

ZenTune can perform an analysis one or more times each day, depending on the values of these two options:

- `tune-offset` *Value* -- Sets the number of minutes after midnight when the ZenTune will first run. By default, the value is 0.
- `tune-interval` *Value* -- Sets the number of minutes to wait before running ZenTune again. By default, the value is 0, which is equivalent to 1440 (24 hours).

So, for example, if you want ZenTune to run twice each day, set the value of tune-offset to 0 and the value of tune-interval to 720.

Configuring ZenTune for Remote Databases

If you have installed the Zenoss DataStore on a server other than your master server, then you must set additional configuration options. Set the following options in the `$ZENHOME/etc/zentune.conf` file of the master server:

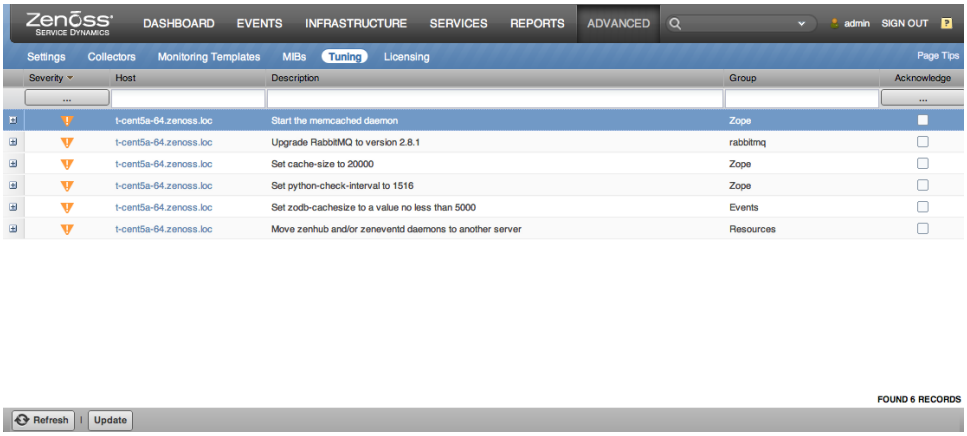
Table 6: Remote Database Configuration Options

Option	Description
`mysqltuner-zodb-forcemem` *MegaBytes*	Sets the amount of memory available on the server running the ZODB database server.
`mysqltuner-zep-forcemem` *MegaBytes*	Sets the amount of memory available on the server running the ZEP database server.

Using ZenTune

To access ZenTune, select Advanced > Tuning from the Resource Manager interface.

Figure 4: ZenTune

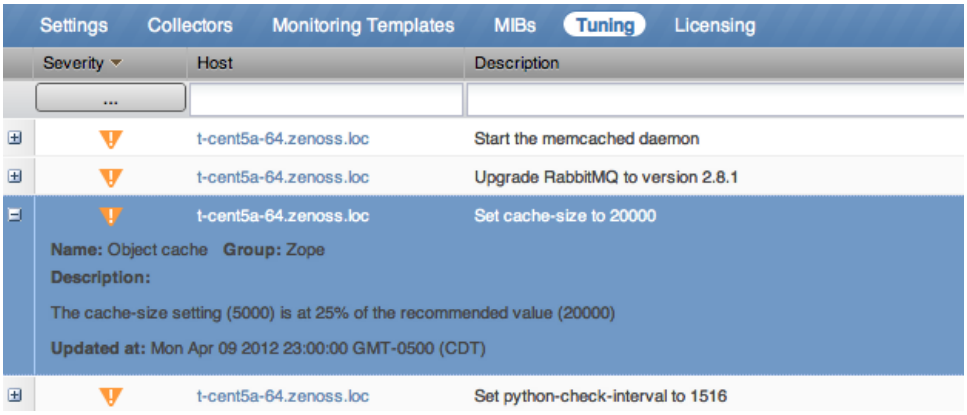


To run ZenTune, click **Update** (located at the bottom left of the page). ZenTune may require several minutes to run.

Note To check the update status, refresh the browser page and then check the "Update at" value for any watched items.

ZenTune returns information about current and optimal values for several configuration parameters. Click + to the left of each item to display recommendations, if any, for configuration changes.

Figure 5: ZenTune Issue Detail



To refresh the view, click **Refresh**. (This does not run ZenTune again.)

To filter the list of displayed items, select Not Acknowledged, Acknowledged, or both in the Acknowledge column. To acknowledge one or more items, select the option in the Acknowledge column.

You also can filter the display by severity, host, and description.

Running ZenTune from the Command Line

You can run ZenTune from the command line. On the master server, use the command:

```
zentune run
```

To run ZenTune on a remote hub or collector, prefix the command with the name of the hub or collector, followed by an underscore. For example, if the remote collector name is centos6-coll, then run the command as:

```
centos6-coll_zentune run
```

This generates a report to the console output. If you additionally specify the `--events` option, events are instead issued (the same events issued by the `zentune` daemon). The results appear on the Tuning page of the Resource Manager interface.

Tuneable Items

The following table lists included tuneable items.

Name	Group	Description
zeneventd workers	Events	ZenTune monitors the number of incoming events to the `zeneventd` daemon. If the number of incoming events exceeds the configured threshold per worker, then a tuning event is generated. If the threshold is exceeded only during peak load times, a WARNING severity tuning event is generated. If it is exceeded more often, an ERROR severity tuning event is generated. The threshold checked is controlled by these configuration options on the `zentune` daemon:

Name	Group	Description
		■ zeneventd-worker-events-per-second ■ zeneventd-worker-count-max-recommended For detailed information about each of these options, run this command on any Resource Manager server: `zentune run --help`
zeneventserver	Events	
zeneventserver age-eligible events	Events	
zeneventserver archive-eligible events	Events	
zeneventserver processed events	Events	
zeneventserver deduped events	Events	
zeneventserver dropped events	Events	
zeneventserver cleared events	Events	
zeneventserver archived events	Events	
zeneventserver aged events	Events	
zeneventserver summary queue length	Events	
zeneventserver archive queue length	Events	
zeneventserver summary index size	Events	
zeneventserver archive index size	Events	
zeneventserver summary index doc count	Events	
zeneventserver archive index doc count	Events	
age eligible event count	Events	
archive eligible event count	Events	

Name	Group	Description
zeneventserver summary queue length	Events	
zeneventserver archive queue length	Events	
zeneventd object cache	Global	The zeneventd zodb-cachesize configuration setting controls the number of objects that zeneventd will store locally to avoid querying ZODB. Resource Manager expects this value to be between certain thresholds, proportional to the size of the global catalog. If any of these thresholds are violated, then a WARNING or ERROR severity tuning event is generated, depending on the configured thresholds. The thresholds checked are controlled by the following configuration options on the `zentune` daemon: ■ zeneventd-obj-cache-bad ■ zeneventd-obj-cache-warn For detailed information about each of these options, run this command on any Resource Manager server: `zentune run --help`
ZODB cache servers	Global	Resource Manager expects at least one memcached server to be configured for its use as a ZODB cache. If this is not the case, then an ERROR severity tuning event is generated. Memcached is a third-party object caching system used by Resource Manager to improve performance for daemons that connect to Zope and ZODB. It is not required for Resource Manager to function, but it is highly recommended. More information on memcached can be found here: *http://memcached.org/*
globalConfig	Global	
Global config sip size	Global	The configsipsize global configuration setting controls the number of device configuration objects that a collector daemon will receive in a single request. A setting of 0 implies that all device configurations will be requested at once. Combined with the configsipdelay option, this effectively controls the traffic to collector daemons.
Global config sip delay	Global	The configsipdelay global configuration setting controls the number of seconds between requests for device configuration objects that a collector daemon will make. Combined with the configsipsize option, this effectively controls the traffic to collector daemons.

Name	Group	Description
Event flush chunk size	Global	The eventflushchunksize global configuration setting controls the number of events a collector daemon will send to zenhub at one time. Each collector daemon will periodically flush its outgoing event queue and send events to zenhub until the queue is empty.
Maximum queue length	Global	The maxqueuelen global configuration setting controls the number of events a collector daemon can store in its outgoing event queue before it must start dropping events.
IO CPU Wait Time	IO	ZenTune monitors iostat statistics, and if any device causes wait times exceeding the configured threshold in more than 5% of cases, then an ERROR severity tuning event is generated. The threshold checked is controlled by the following configuration option on the `zentune` daemon: ■ iostat-wait-threshold For detailed information about each of these options, run this command on any Resource Manager server: `zentune run --help` The iostat utility is a third-party program that provides statistics on the time the CPU spends waiting for I/O requests from various devices. More information on iostat can be found here: *http://en.wikipedia.org/wiki/Iostat*
Cache miss percentage	Memcached	Memcached is a third-party object caching system used by Resource Manager to improve performance for daemons that connect to Zope and ZODB. It is not required for Resource Manager to function, but it is highly recommended. More information on memcached can be found here: *http://memcached.org/*
Maximum size	Memcached	Memcached is a third-party object caching system used by Resource Manager to improve performance for daemons that connect to Zope and ZODB. It is not required for Resource Manager to function, but it is highly recommended. More information on memcached can be found here: *http://memcached.org/*
Cache eviction rate	Memcached	Memcached is a third-party object caching system used by Resource Manager to improve performance for daemons that connect to Zope and ZODB. It is not required for Resource Manager to function, but it is highly recommended. More information on memcached can be found here:

Name	Group	Description
		http://memcached.org/
Cache servers	Memcached	Resource Manager expects at least one memcached server to be configured for its use. If this is not the case, then an INFO severity tuning event is generated. Resource Manager also expects all configured memcached servers to be available and responding to connection attempts using the standard memcached client. If this is not the case, then an ERROR severity tuning event is generated. Memcached is a third-party object caching system used by Resource Manager to improve performance for daemons that connect to Zope and ZODB. It is not required for Resource Manager to function, but it is highly recommended. More information on memcached can be found here: *http://memcached.org/*
Cache size	Memcached	Resource Manager expects the use of each memcached server to conform to certain performance thresholds. Specifically, the utilization level and eviction rate of each memcached server are checked. If any of these thresholds are violated, then a WARNING or ERROR severity tuning event is generated, depending on the configured thresholds. The thresholds checked are controlled by the following configuration options on the `zentune` daemon: ■ memcache-size-high-warn ■ memcache-size-high-bad ■ memcache-size-low-warn ■ memcache-size-low-bad ■ memcache-size-evictions-warn ■ memcache-size-evictions-bad For detailed information about each of these options, run this command on any Resource Manager server: `zentune run --help` Memcached is a third-party object caching system used by Resource Manager to improve performance for daemons that connect to Zope and ZODB. It is not required for Resource Manager to function, but it is highly recommended. More information on memcached can be found here: *http://memcached.org/*
MySqlTuner script	Resources	ZenTune expects the `mysqltuner.pl` utility to be installed and available for its use to enable more detailed tuning advice. The `mysqltuner.pl` utility is a third-party tuning script that provides advanced statistics on MySQL. More information on `mysqltuner.pl` can be found here:

Name	Group	Description
		http://mysqltuner.pl/help MySQL is a third-party, open-source relational database. Resource Manager uses MySQL as the backing data store for ZODB, as well as directly to store events and user sessions. More information on MySQL can be found here: *http://www.mysql.com/*
iostat Utility	Resources	ZenTune expects the iostat utility to be installed and available for its use to enable more detailed tuning advice. The iostat utility is a third-party program that provides statistics on the time the CPU spends waiting for I/O requests from various devices. More information on iostat can be found here: *http://en.wikipedia.org/wiki/Iostat*
Memory	Resources	This tuning item provides information about the total amount of RAM installed in the Resource Manager master server.
Processes	Resources	This tuning item provides advice on the distribution of CPU-intensive Resource Manager processes according to the number of cores available on the Resource Manager server. If any of the thresholds are violated, a WARNING or ERROR severity tuning event will be generated, depending on the configured thresholds. The thresholds checked are controlled by the following configuration options on the zentune daemon: ■ resources-available-cores-warn ■ resources-available-cores-bad For detailed information about each of these options, run this command on any Resource Manager server: `zentune run --help`
MySQL Version	MySQL Database	Resource Manager expects that at least a specific, minimum version of MySQL is installed and available for its use. Earlier versions may not support all the features that Resource Manager requires, or may have hidden incompatibilities. If this minimum threshold is violated, then an ERROR severity tuning event is generated, depending on the configured threshold. The threshold checked is controlled by the following configuration option on the zentune daemon: ■ mysql-recommended-version For detailed information about this option, run this command on any Resource Manager server: `zentune run --help` MySQL is a third-party, open-source relational database. Resource Manager uses MySQL as the backing data store for

Name	Group	Description
		ZODB, as well as directly to store events and user sessions. More information on MySQL can be found here: *http://www.mysql.com/*
InnoDB buffer pool size	MySQL Database	If the `mysqltuner.pl` script recommends increasing the amount of memory available to InnoDB, then an ERROR level tuning event is generated. More information on the `innodb_buffer_pool_size` configuration setting can be found here: *http://dev.mysql.com/doc/refman/5.5/en/innodb-parameters.html\#sysvar_innodb_buffer_pool_size* MySQL is a third-party, open-source relational database. Resource Manager uses MySQL as the backing data store for ZODB, as well as directly to store events and user sessions. More information on MySQL can be found here: *http://www.mysql.com/*
Table sizes	MySQL Database	This tuning item provides information about the total size and number of tables in each MySQL instance configured for Resource Manager. MySQL is a third-party open-source relational database. Zenoss uses MySQL as the backing data store for ZODB, as well as directly to store events and user sessions. More information on MySQL can be found here: http://www.mysql.com/ MySQL is a third-party, open-source relational database. Resource Manager uses MySQL as the backing data store for ZODB, as well as directly to store events and user sessions. More information on MySQL can be found here: *http://www.mysql.com/*
Table fragmentation	MySQL Database	If the `mysqltuner.pl` script recommends de-fragmenting the tables in a MySQL instance, then an ERROR level tuning event is generated. MySQL is a third-party, open-source relational database. Resource Manager uses MySQL as the backing data store for ZODB, as well as directly to store events and user sessions. More information on MySQL can be found here: *http://www.mysql.com/*
Thread cache	MySQL Database	If the `mysqltuner.pl` script recommends increasing the number of threads cached for reuse by MySQL, then an ERROR level tuning event will be generated. More information on the `thread_cache_size` configuration setting can be found here: *http://dev.mysql.com/doc/refman/5.5/en/server-system-variables.html#sysvar_thread_cache_size*

Name	Group	Description
		MySQL is a third-party, open-source relational database. Resource Manager uses MySQL as the backing data store for ZODB, as well as directly to store events and user sessions. More information on MySQL can be found here: *http://www.mysql.com/*
Version	rabbitmq	
Hub	Hubs	
zenhub workers	Hubs	
Check interval	Zope	
Object cache	Zope	
RelStorage cache	Zope	
Pool size	Zope	
Cache servers	Zope	
Maximum number of session objects	Zope	
Debug mode	Zope	
Application server processes	Zope	
Application server threads	Zope	
Request latency	Zope	

Daemons

Type	Name
Performance Collector	`zentune`

(BigIpMonitor) BIG-IP Devices 6

The ZenPacks.zenoss.BigIpMonitor ZenPack monitors BIG-IP devices (from F5 Networks).

The following metrics are collected:

- CPU utilization
- memory utilization
- per-instance metrics for each load-balanced virtual server that is configured

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x
Required ZenPacks	ZenPacks.zenoss.BigIPMonitor

Enable Monitoring

To add a device and enable BIG-IP monitoring on it:

1 From Infrastructure, select Add a Single Device from Add Device.

The Add a Single Device page appears.

2 Enter a name for the device, and then select these values:

- **Model Device** - De-select this option.
- **Device Class** - Select `/Network/BIG-IP`.

3 Click **Add**.

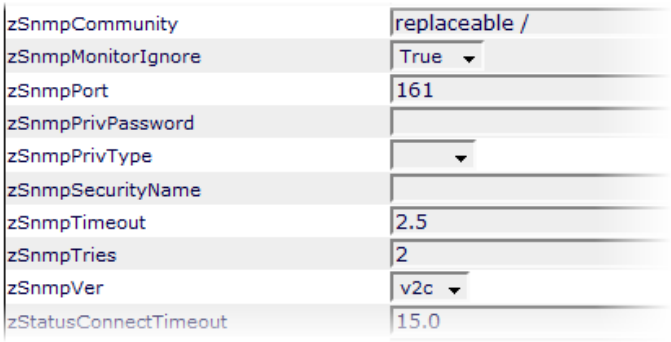
4 Navigate to the newly created device.

5 Select Configuration Properties in the left panel.

6 Change the values of these configuration properties:

- **zSnmpCommunity** - Enter the SNMP community string here.
- **zSnmpVer** - Select `v2c`.

Figure 6: BIG-IP Configuration Properties Selections



7 Click **Save**.

8 Model the device. To to this, select Manage > Model Device from the page menu.

Resource Manager models the device. When modeling completes, you can view the device. After approximately fifteen minutes, you can verify that the performance graphs are updating.

Viewing Virtual Servers

To view the virtual servers, select BIG-IP details. Click a link in the table to view additional information for each load-balanced server.

Daemons

Type	Name
Modeler	`zenmodeler`
Performance Collector	`zenperfsnmp`

(BrocadeMonitor) Brocade SAN Switches 7

The ZenPacks.zenoss.BrocadeMonitor ZenPack monitors Brocade Storage Area Network (SAN) switches.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x
Required ZenPacks	ZenPacks.zenoss.BrocadeMonitor, ZenPacks.zenoss.StorageBase

Configuring Brocade Devices to Allow SNMP Queries

Configure the Brocade devices to allow SNMP queries from the Resource Manager server, and send SNMP v1 or SNMP v2 traps to the Resource Manager server.

Configuring Resource Manager

All Brocade devices must exist under the `/Devices/Storage/Brocade` device class.

1. Navigate to the device or device class in the Resource Manager interface.
   - If applying changes to a device class:
     1. Select the class in the devices hierarchy.
     2. Click **Details**.
     3. Select Configuration Properties.
   - If applying changes to a device:
     1. Click the device in the device list.
     2. Select Configuration Properties.
2. Edit the appropriate configuration properties for the device or devices.

Table 7: Brocade Configuration Properties

Name	Description
zSnmpCommunity	Consult with your storage administrators to determine the SNMP community permitted
zSnmpMonitorIgnore	This should be set to `False`
zSnmpPort	The default port is 161
zSnmpVer	This should be set to `v2c`

3 Click **Save** to save your changes. You will now be able to start collecting the Brocade switch metrics from this device.

Viewing Fibre Channel Port Information

To view the virtual servers, select Brocade Details.

Daemons

Type	Name
Modeler	`zenmodeler`
Performance Collector	`zenperfsnmp`

(CheckPointMonitor) Check Point Security Appliance 8

The ZenPacks.zenoss.CheckPointMonitor ZenPack monitors security appliances from Check Point.

With this ZenPack, you can ensure that the firewall module has a policy installed, HA is in a proper state, and that the policy server (for SecureClient) is running.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x
Required ZenPacks	ZenPacks.zenoss.CheckPointMonitor

Configuring Check Point Firewalls to Allow SNMP Queries

Configure the Check Point firewall to allow SNMP queries from Resource Manager, and to send SNMP v1 or SNMP v2 traps to Resource Manager.

Configuring Resource Manager

All Check Point devices must exist under the `/Devices/Network/Check Point` device class.

1. Navigate to the device or device class in the Resource Manager interface.
   - If applying changes to a device class:
     1. Select the class in the devices hierarchy.
     2. Click **Details**.
     3. Select Configuration Properties.
   - If applying changes to a device:
     1. Click the device in the device list.
     2. Select Configuration Properties.
2. Edit the appropriate configuration properties for the device or devices.

Table 8: Check Point Configuration Properties

Name	Description
zSnmpCommunity	Consult with your network administrators to determine the SNMP community permitted.
zSnmpMonitorIgnore	This should be set to `False`
zSnmpPort	The default port is 161
zSnmpVer	This should be set to `v2c`

3 Click **Save** to save your changes.

You will now be able to start collecting the Check Point firewall metrics from this device.

4 Navigate to Graphs and you should see some placeholders for performance graphs. After approximately fifteen minutes you should see the graphs start to become populated with information.

Daemons

Type	Name
Modeler	zenmodeler
Performance Collector	zenperfsnmp

(CiscoMonitor) Cisco Devices 9

The ZenPacks.zenoss.CiscoMonitor ZenPack monitors faults and performance of a wide range of Cisco equipment, including virtual resources such as virtual firewalls and virtual load balancers.

Monitored devices

- Cisco ASA 5500 Series Adaptive Security Appliance
- Cisco ASR 9000 Series Aggregation Services Routers
- Cisco Nexus 7000 Series Switches
- Cisco Catalyst 6500 Series Switches
- Cisco ASR 1000 Series Aggregation Services Routers
- Cisco Nexus 5000 Series Switches
- Cisco Catalyst 6500 Series Virtual Switching Systems (VSS)
- Cisco MDS 9000 Series Multilayer Switches
- Cisco Nexus 2000 Series Fabric Extenders
- Cisco Application Control Engine (ACE) Modules for Catalyst 6500 Series
- Cisco Wireless LAN Controllers (WLC)
- Cisco Nexus 1000v Series Switches
- Cisco Firewall Services Modules (FWSM) for Catalyst 6500 Series
- Cisco TelePresence Codecs
- Cisco Virtual Security Gateway (VSG) for Nexus 1000v Series Switches

Additional monitoring

Additional monitoring is supported for the following Cisco product lines.

- Catalyst 6500 Series Switches
- Catalyst 6500 Series Virtual Switching Systems (VSS)
- Application Control Engine (ACE) Modules for Catalyst 6500 Series
- Firewall Services Modules (FWSM) for Catalyst 6500 Series
- ASA Services Modules (ASA-SM) for Catalyst 6500 Series
- ASA 5500 Series Adaptive Security Appliances
- ASA 1000V Cloud Firewalls
- Nexus 7000 Series Switches
- Nexus 5000 Series Switches
- Nexus 3000 Series Switches

- Nexus 2000 Series Fabric Extenders
- Nexus 1010 Series Virtual Services Appliances
- Nexus 1000v Series Switches
- Virtual Security Gateway (VSG) for Nexus 1000v Series Switches
- ASR 9000 Series Aggregation Services Routers
- ASR 1000 Series Aggregation Services Routers
- CSR 1000V Cloud Services Routers
- MDS 9000 Series Multilayer Switches
- Wireless LAN Controllers (WLC)
- TelePresence Codecs

Supported common features

The following common features are available across the supported products (where available).

Base Discovery	■ Chassis ■ Supervisor Modules ■ Line Cards ■ Power Supplies ■ Fans ■ Temperature Sensors ■ Physical Ports and Interfaces ■ Port Channels and Bundles ■ Other Logical Interfaces ■ VLANs ■ VRFs ■ QoS Class Maps
Base Monitoring	■ Event collection from syslog and SNMP traps ■ CPU and memory utilization for chassis and supervisor modules ■ Power consumption and status for chassis and FRUs ■ Power available and drawn for power supplies ■ Temperature for temperature sensors ■ Interface utilization, throughput, error rate and status for all physical Ethernet interfaces ■ Interface utilization, throughput and status for all logical Ethernet interfaces ■ Throughput and status for VLANs

Supported discovery and monitoring

This ZenPacks supports discovery and monitoring of the product line features listed in the following table.

Product line	Features discovered and monitored
Catalyst 6500	Virtual switching system (VSS)
	Service modules
ACE	Virtual contexts
	Service policies

Product line	Features discovered and monitored
	Server farms
	Real servers
FWSM	Security contexts
	L4/L7 resources
Nexus 7000	Fabric cards
	Virtual device contexts (VDCs)
Nexus 5000	Nexus 2000 fabric extenders
	Fibre-channel ports
	VSANs
	Storage zones
	Storage zone sets
Nexus 1010	Virtual services blades
Nexus 1000V	Virtual Ethernet modules (VEMs)
	Virtual Ethernet interfaces
VSG	Virtual machine zones
Virtual Security Gateway	Security zones
ASR 9000 and 1000	MPLS L3 VPNs
CSR 1000V	Similar to ASR 1000 without hardware components
ASA 5500	Security contexts
ASA 1000V	Similar to ASA 5500 without hardware components
MDS 9000	Fibre-channel ports
	VSANs
	Storage zones
	Storage zone sets
Wireless LAN Controller	Access points
TelePresence Codecs	Telepresence peripherals
Quality of Service (QoS) Class Maps	QoS traffic shaping

Configuring

Configuring this ZenPack includes ensuring monitored devices are placed in the correct device classes, and providing credentials for the network protocols used in monitoring.

Devices must be placed in the correct device classes to ensure reliable discovery and monitoring. The discovery feature automatically assigns a Cisco device to its device class based on the device's `sysObjectID`. This allows

you to initiate discovery of a single Cisco device from the `/Discovered` device class, or to discover an entire subnet and have all Cisco devices correctly classified.

Resource Manager uses different network protocols to monitor Cisco devices, and in many cases, uses multiple protocols for the same device. The credentials to enable these protocols must be configured.

Device Class Mappings

Cisco device type	Resource Manager device class
Catalyst 6500	/Network/Cisco/6500
Catalyst 6500 VSS	/Network/Cisco/6500/VSS
ACE	/Network/Cisco/ACE
FWSM	/Network/Cisco/FWSM
Nexus 7000	/Network/Cisco/Nexus/7000
Nexus 5000, Nexus 2000	/Network/Cisco/Nexus/5000
Nexus 3000	/Network/Cisco/Nexus/3000
Nexus 1010	/Network/Cisco/Nexus/1010
Nexus 1000V	/Network/Cisco/Nexus/1000V
VSG	/Network/Cisco/VSG
ASR 9000	/Network/Cisco/ASR/9000
ASR 1000	/Network/Cisco/ASR/1000
ASA	/Network/Cisco/ASA
ASA-SM	/Network/Cisco/ASA
MDS 9000	/Network/Cisco/MDS/9000
WLC	/Network/Cisco/WLC
TelePresence Codec	/Network/Cisco/Codec
Other IOS	/Network/Cisco
Other CatOS	/Network/Cisco/CatOS

Discovery and Monitoring Protocols

Cisco device type	Discovery and monitoring protocols
Catalyst 6500	ICMP, SNMP, Telnet or SSH
Catalyst 6500 VSS	ICMP, SMP, Telnet or SSH
ACE	ICMP, SNMP, API (XML over HTTP: 80/TCP)
FWSM	ICMP, SNMP
Nexus 7000	ICMP, SNMP, Netconf (XML over SSH)
Nexus 5000, Nexus 2000	ICMP, SNMP, Netconf (XML over SSH)
Nexus 3000	ICMP, SNMP, Netconf (XML over SSH)

Cisco device type	Discovery and monitoring protocols
Nexus 1010	ICMP, SNMP, Netconf (XML over SSH)
Nexus 1000v	ICMP, SNMP, Netconf (XML over SSH)
VSG	ICMP, SNMP, SSH, Netconf (XML over SSH)
ASR 9000	ICMP, SNMP, Telnet or SSH
ASR 1000	ICMP, SNMP
ASA	ICMP, SNMP
ASA-SM	ICMP, SNMP
MDS 9000	ICMP, SNMP
WLC	ICMP, SNMP
TelePresence Codec	ICMP, SNMP
Other IOS	ICMP, SNMP
Other CatOS	ICMP, SNMP

Configuration Properties Settings

Protocol	Configuration properties	Notes
SNMPv1, SNMPv2c	zSnmpCommunities, zSnmpCommunity	SNMPv1 is only used if SNMPv2c tests fail.
SNMPv3	zSnmpSecurityName, zSnmpAuthType, zSnmpAuthPassword, zSnmpPrivType, zSnmpPrivPassword	SNMPv3 must be specified using zSnmpVer.
Telnet, SSH, Netconf	zCommandProtocol, zCommandPort, zCommandUsername, zCommandPassword	For SSH, set zCommandProtocol to `ssh` and zCommandPort to `22`. For Telnet, set zCommandProtocol to `telnet` and zCommandPort to `23`.
ACE XML over HTTP API	zCommandUsername, zCommandPassword	

Prerequisites

Prerequisite	Restriction
Zenoss platform	Resource Manager 4.x
Required ZenPacks	ZenPacks.zenoss.CiscoMonitor 4.0 or the most recent version
Resource Manager daemons	zenmodeler, zenperfsnmp, zencommand, zentrap, zensyslog

Firewall Access

Resource Manager requires firewall access between collector servers and monitored devices.

Source	Destination	Port and Protocol
Resource Manager collector	Monitored device	ICMP (Ping)
Resource Manager collector	Monitored device	161/UDP (SNMP)
Resource Manager collector	Monitored device	22/TCP (SSH)
Resource Manager collector	Monitored device	23/TCP (Telnet)
Resource Manager collector	Monitored device	80/TCP (HTTP)
Monitored device	Resource Manager collector	ICMP (Ping)
Monitored device	Resource Manager collector	162/UDP (SNMP trap)
Monitored device	Resource Manager collector	514/UDP (syslog)

Limitations

Currently, this ZenPack has the following limitations:

- Cisco UCS is not supported. Install the ZenPacks.zenoss.CiscoUCS ZenPack to enable UCS support.
- Cisco CallManager is not supported. Install the ZenPacks.zenoss.CallManagerMonitor ZenPack to enable UCS support.

Monitoring Logical Contexts

Several of the supported device types can create various kinds of logical contexts. Resource Manager identifies the logical contexts and associates them with an admin or parent context. The following list associates Cisco device types with the terms Resource Manager uses to refer to their logical contexts.

ACE

Virtual Contexts

FWSM

Security Contexts

ASA

Security Contexts

Nexus 7000

VDCs

For Resource Manager to discover and associate these logical contexts with an admin or parent context, it must also be able to discover the management IP address of each logical context as a separate device. Logical contexts are placed in the same device class as the device they represent. For example, a Nexus 7000 VDC is placed in the /Network/Cisco/Nexus/7000 device class.

Controlling automatic remodeling

Certain SNMP traps cause Resource Manager to schedule an immediate remodeling of the device from which the trap was sent. The following traps, defined in the zCiscoRemodelEventClassKeys configuration property, cause automatic remodeling.

- ccmCLIRunningConfigChanged
- cefcFRUInserted
- cefcFRURemoved
- cefcModuleOperStatus

- cefcModuleStatusChange
- ceSensorExtThresholdNotification
- cesRealServerStateChangeRev1
- ciscoSlbExtMIBConform
- ciscoSlbVServerStateChange
- entConfigChange
- vtpVlanCreated
- vtpVlanDeleted

Monitoring QoS

By default, QoS class maps are only discovered and monitored for devices in the /Network/Cisco/MPLS device class. If you have devices that belong in one of the other /Network/Cisco/* devices or device classes on which you also want to monitor QoS, you can add the cisco.snmp.QoSClassMaps modeler plugin.

Installed MIBs

This ZenPack installs the following MIBs. Any SNMP traps defined in these MIBs are decoded by Resource Manager.

- BRIDGE-MIB
- CISCO-CONFIG-MAN-MIB
- CISCO-ENHANCED-SLB-MIB
- CISCO-ENTITY-FRU-CONTROL-MIB
- CISCO-ENTITY-SENSOR-EXT-MIB
- CISCO-ENTITY-SENSOR-MIB
- CISCO-ENVMON-MIB
- CISCO-FC-FE-MIB
- CISCO-FLEX-LINKS-MIB
- CISCO-HSRP-MIB
- CISCO-IF-EXTENSION-MIB
- CISCO-L2-TUNNEL-CONFIG-MIB
- CISCO-LINK-ERROR-MONITOR-MIB
- CISCO-MAC-NOTIFICATION-MIB
- CISCO-MODULE-AUTO-SHUTDOWN-MIB
- CISCO-NS-MIB
- CISCO-OSPF-MIB
- CISCO-OSPF-TRAP-MIB
- CISCO-SLB-EXT-MIB
- CISCO-SLB-HEALTH-MON-MIB
- CISCO-SLB-MIB
- CISCO-SMI
- CISCO-ST-TC
- CISCO-STP-EXTENSIONS-MIB
- CISCO-TC
- CISCO-VLAN-MEMBERSHIP-MIB
- CISCO-VSAN-MIB
- CISCO-VTP-MIB
- CISCO-ZS-MIB
- ENTITY-MIB

- ENTITY-SENSOR-MIB
- HC-ALARM-MIB
- IF-MIB
- OLD-CISCO-INTERFACES-MIB
- OLD-CISCO-SYSTEM-MIB
- OLD-CISCO-TCP-MIB
- OLD-CISCO-TS-MIB
- P-BRIDGE-MIB
- Q-BRIDGE-MIB
- RMON2-MIB
- SNMPv2-MIB
- TOKEN-RING-RMON-MIB

(CiscoUCS) Cisco UCS 10

The ZenPacks.zenoss.CiscoUCS ZenPack uses HTTP to monitor Cisco Unified Computing System (UCS) devices.

Resource Manager uses Cisco's UCS™ Manager XML API to model and monitor devices placed in the `/CiscoUCS` device class.

This ZenPack provides:

- Initial discovery and periodic remodeling of components
- Performance monitoring
- Event management
- Integration with Zenoss Service Dynamics Service Impact

Discovery

The following components are automatically discovered through the Cisco UCS Manager host, user and password you provide. The properties and relationships are periodically remodeled to provide up-to-date monitoring when the system configuration changes.

Component	Properties	Relationships
UCS Manager (Device)	None	Service Profiles, Management Interfaces, Fabric Interconnects, Chassis
Service Profiles	DN, Type, Description	UCS Manager, Virtual Ethernet NICs, Bound Server
Virtual Ethernet NICs	DN, MAC Address	Service Profile, Bound Host Ethernet Interface
Management Interfaces	Name, Type, MAC Address, Administrative Status, Operational Status	UCS Manager, IP Addresses
Fabric Interconnects	DN, Manufacturer, Model, Serial Number, Total Memory	UCS Manager, Switch Cards, Power Supply Units
Switch Cards	DN, Description, Revision, Slot, Manufacturer, Model	Fabric Interconnect, Ethernet Ports, Fibre Channel Ports

Component	Properties	Relationships
Ethernet Ports	DN, Switch ID, Slot ID, Port ID, MAC Address, Role, Type, Transport, Administrative State, Endpoint DN, Peer DN, Peer Slot ID, Peer Port ID	Switch Card
Fibre Channel Ports	DN, WWN, Operational Speed	Switch Card
Chassis	DN, Manufacturer, Model, Serial Number, Operational State	UCS Manager, Servers, Fan Modules, Power Supply Units
Power Supply Units	DN, Manufacturer, Model, Serial Number, Revision, Performance Threshold Sensor Status, Power State, Thermal Threshold Sensor Status, Voltage Threshold Sensor State	Equipment
Fan Modules	DN, Manufacturer, Model, Serial Number, Tray	Chassis
Server Blades	DN, Manufacturer, Model, Serial Number, Presence, Operational State, Operational Power, Availability	Chassis, Memory Arrays, Processor Units, Adaptor Units, Bound Service Profile
Adaptor Units	DN, Manufacturer, Model, Serial Number	Server Blade, Host Ethernet Interfaces
Processor Units	DN, Manufacturer, Model, Serial Number, Revision, Socket, Architecture, Cores, Threads, Stepping, Speed, Voltage	Server Blade
Memory Arrays	DN, Serial Number, Revision, Error Correction, CPU ID, Max Devices, Populated Devices, Max Capacity, Current Capacity	Server Blade
Host Ethernet Interfaces	DN, Manufacturer, Model, MAC Address, PCI Address	Adaptor Unit

Performance monitoring

The following metrics are collected every 5 minutes.

Component	Metric		
	Category	Units	Name
Chassis	Power	watts	Input, Output
Ethernet Ports	Throughput	bits/sec	Sent, Received
	Sent Packets	packets/sec	Total, Jumbo, Unicast, Broadcast, Multicast
	Received Packets	packets/sec	Total, Jumbo, Unicast, Broadcast, Multicast
	Loss Stats	losses/sec	Carrier Sense, Excess Collision, Giants, Late Collision, Multi Collision, Single Collision
	Pause Stats	pauses/sec	Transmit, Receive, Resets
	Errors	errors/sec	Transmit, Receive, Deferred Tx, Out Discard, Under Size, Align, FCS, Int Mac Rx, Int Mac Tx

Component	Metric		
	Category	**Units**	**Name**
Fabric Interconnects	CPU Utilization	percent	Used
	Memory Utilization	bytes	Available, Cached
	Temperature	degrees C	PSU Inlet 1, PSU Inlet 2, Fan Inlet 1, Fan Inlet 2, Fan Inlet 3, Fan Inlet 4, Main Outlet 1, Main Outlet 2
Fan Modules	Temperature	degrees C	Ambient
	Fan Speeds	RPM	Fan 1, Fan 2
Fibre Channel Ports	Throughput	bits/sec	Sent, Received
	Errors	errors/sec	Transmit, Receive, Discard Tx, Discard Rx, Too Long Rx, Too Short Rx, CRC Rx, Link Failures, Signal Losses, Sync Losses
Host Ethernet Interfaces	Sent Packets	packets/sec	Total, Good, VLAN, PPP, Pause, Priority Pause
	Received Packets	packets/sec	Total, Good, VLAN, PPP, Pause, Priority Pause
Power Supply Units	Voltages	volts	210V Input, 12V Output, 3V3 Output
	Temperature	degrees C	Ambient
	Power	watts	Output
	Current	amps	Output
Processor Units	Temperature	degrees C	Processor
	Current	amps	Input
Server Blades	Voltage	volts	Input
	Temperature	degrees C	IO, Rear
	Power	watts	Consumed
	Current	amps	Input

Event Management

Resource Manager creates events for all UCS faults it collects. The UCS fault life-cycle closely matches that of the Zenoss event life-cycle, so when a UCS fault clears, the equivalent events are cleared in Resource Manager.

Upon initial connection to UCS Manager, Resource Manager processes the full list of open faults. Subsequently it subscribes to, and only receives, new faults and updates to existing faults. Initial connections to UCS Manager occur when UCS or Resource Manager restarts, and after a temporary connectivity issue between the two is resolved.

The following, standard event fields are populated for each event.

- Device (set to the UCS Manager device in the /CiscoUCS device class)
- component
- eventKey
- summary
- message

- severity
- eventClassKey
- agent (**zenucsevents**)
- eventGroup (ucs)
- monitor

In addition, the following, customized event fields are populated for each event.

- ucs-code
- ucs-id
- user
- originaltime

Integration with Service Impact

This ZenPack includes custom state providers for services running on UCS Manager. Custom state providers enable specialized options for defining state triggers in Zenoss Service Dynamics Service Impact (Service Impact).

The following relationships are automatically included in any Service Impact services that contain one or more of the explicitly mentioned components.

- UCS Manager failure impacts related service profiles and fabric interconnects
- Service profile failure impacts guest operating system
- Virtual ethernet NIC failure impacts related service profile
- Fabric interconnect failure impacts related switch cards and power supply units
- Switch card failure impacts related ethernet and fibre channel ports
- Chassis failure impacts related server blades
- Power supply unit failure impacts related chassis
- Fan failure impacts related chassis
- Server blade failure impacts bound service profile
- Adaptor unit failure impacts related host ethernet interfaces
- Processor unit failure impacts related server blade
- Memory array failure impacts related server blade

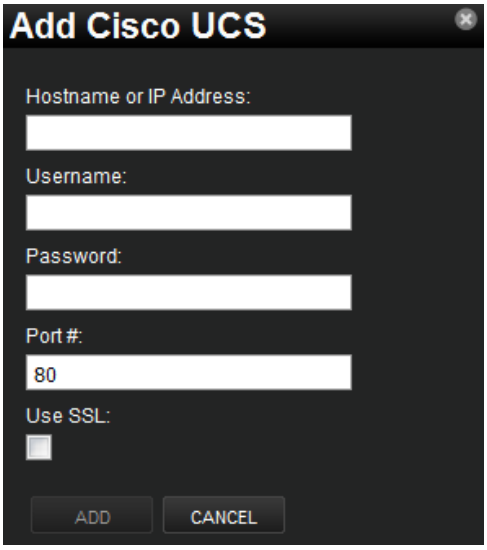
The preceding relationships follow the default policy, in which a node inherits the worst state of the ancestor nodes to which it is related. For example, a switch card failure implies that all related ports are also failed.

Adding a Cisco UCS Device

Follow these steps to begin monitoring a Cisco UCS device through Resource Manager:

1. In the Resource Manager interface, navigate to the `/CiscoUCS` device class.
2. From Add Devices, select Add Cisco UCS.

Figure 7: Add Cisco UCS dialog



3 Enter information in the dialog:

- **Hostname or IP Address** - Enter the host name or IP address of the UCS Manager. Use the floating IP address of the UCS Manager, not the physical IP address of either fabric interconnect.
- **Username** - Enter the username of a user with at least read-only permission to UCS Manager.
- **Password** - Enter the password of the user account.
- **Port #** - By default, Resource Manager assumes a standard HTTP port of 80. Change this value as needed.
- For secure communications between the Resource Manager collector and the UCS Manager, click the SSL checkbox.

4 Click **ADD** to begin discovery.

Adding multiple devices

Follow these steps to add multiple Cisco UCS devices with the `zenbatchload` command.

1 Log in to the Resource Manager master host as `zenoss`.

2 Create a text file with content similar to the following example.

```
/Devices/CiscoUCS loader='ciscoucs', loader_arg_keys=['host',
 'username', 'password', 'port', 'useSsl', 'collector']
ucsm1 host='FQDN-or-IP-address', username='Username',
 password='Password'
```

Replace *FQDN-or-IP-address* with the fully-qualified domain name or IP address of the Cisco UCS device to add, and replace *Username* and *Password* with valid user account information for the device. You may add multiple endpoints under the same /Devices/CiscoUCS section.

3 Load the devices into Resource Manager. Replace *Filename* with the name of the file created in the preceding step.

```
zenbatchload Filename
```

Installed items

Configuration Properties

zCiscoUCSManagerUser

zCiscoUCSManagerPassword
zCiscoUCSManagerPort
zCiscoUCSManagerUseSSL

Device Classes

/CiscoUCS

Modeler Plugins

zenoss.ucs.CiscoUCSInstanceMap

Datasource Types

Cisco UCS XML API

Monitoring Templates (in /CiscoUCS)

Device
UCSChassis
UCSEthPort
UCSFabricInterconnect
UCSFanModule
UCSFcPort
UCSHostEthIf
UCSPowerSupplyUnit
UCSProcessorUnit
UCSServer

Event Classes

/CiscoUCS
/CiscoUCS/Event
/CiscoUFS/Fault
/Status/Blade
/Status/Chassis
/Change/Set/UCS

Event Class Mappings

LOCAL0-3-SYSTEM_MSG (in /CiscoUCS)
UCS Events Default (in /CiscoUCS/Event)
UCS Faults Default (in /CiscoUCS/Fault)

Operational Reports (in Cisco UCS Reports)

Hardware Inventory
Free Slots

Collector Daemons

zenucsevents

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x
Required ZenPacks	ZenPacks.zenoss.CiscoUCS
	ZenPacks.zenoss.DynamicView

Prerequisite	Restriction
Cisco applications	Cisco UCS Manager (1.x, 2.0, 2.1)

Daemons

Type	Name
Modeler	`zenmodeler`
Performance Collector	`zencommand`
Event Monitoring	`zenucsevents`

11 (DellMonitor) Dell Hardware

The ZenPacks.zenoss.DellMonitor ZenPack provides customized modeling of devices running Dell OpenManage agents, and includes identification for proprietary Dell hardware.

The following information is collected through SNMP:

- Hardware Model
- Hardware Serial Number
- Operating System
- CPU Information (socket, speed, cache, voltage)
- PCI Card Information (manufacturer, model)

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.DellMonitor
On each remote device	The Dell OpenManage SNMP Agent is used to gather information about the device.

Enable Enhanced Modeling

To enable modeling:

1. Select Infrastructure from the navigation bar.
2. Click the device name in the device list.

   The device overview page appears.
3. Select Modeler Plugins from the left panel.
4. Click **Add Fields** to reveal the list of available plugins.
5. Select the following plugins from the Available fields list, and then drag them to the Plugins list:

   - DellCPUMap
   - DellDeviceMap

- DellPCIMap

6 Remove the following plugins by clicking on the 'X' button located to the right of the plugin.

- zenoss.snmp.CpuMap
- zenoss.snmp.DeviceMap

7 Click **Save** to save the updates.

8 Remodel the device using these new plugins by selecting Model Device from the Action menu.

Daemons

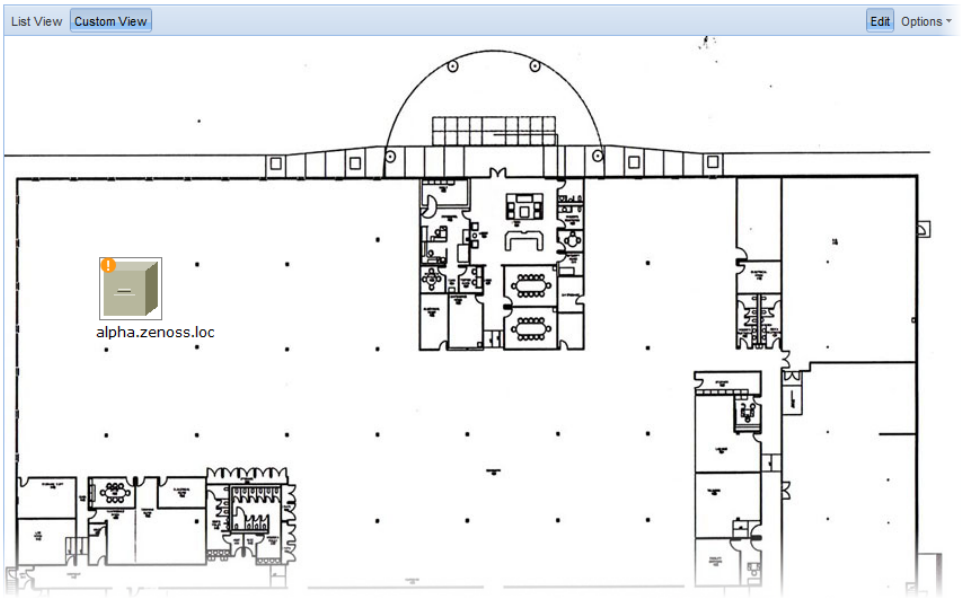
Type	Name
Modeler	zenmodeler
Performance Collector	zenperfsnmp

12 (Diagram) Datacenter View

The ZenPacks.zenoss.Diagram ZenPack enables a visual representation of devices (such as servers or blades) and device containers (such as racks or chassis).

With this feature, you can create a custom view that represents a physical space (such as a data center) by customizing the view background. You can then overlay this view with active representations of your devices and device containers.

Figure 8: Custom View



For each device or device container, the system can generate a rack view, which diagrams the physical location of devices in a chassis or rack. Each represented device provides at-a-glance information about its status.

Figure 9: Rack View

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x
Required ZenPacks	ZenPacks.zenoss.Diagram

Before a device or sub-location can appear in Datacenter View:

- At least one organizer must be configured
- At least one device or sub-organizer must be included in a location

To see the auto-generated rack view, you must set a rack slot value for the device. (For more information, see *Activating the Auto-Generated Rack View*.)

Configuring

Before a device or sub-location can appear in Datacenter View:

- At least one organizer must be configured
- At least one device or sub-organizer must be included in a location

To see the auto-generated rack view, you must set a rack slot value for the device.

Working with the List View

The List View provides a view of your devices (or, if configured, the Rack View).

Follow these steps to access the List View:

1. From the interface, select Infrastructure.
2. In the devices hierarchy, select a location, group, or system.
3. Click **Details**.
4. Select Datacenter View.

   The List View appears.

> **Note** After you create a Custom View, that view appears by default.

Working with the Custom View

The Custom View lets you create a visual representation of your physical space (such as a data center).

To access the Custom View, from the Diagram selection, click Custom View.

You can edit the Custom View to:

- Add or change a background image
- Move or resize device images
- Remove the view

Adding a Background Image to the Custom View

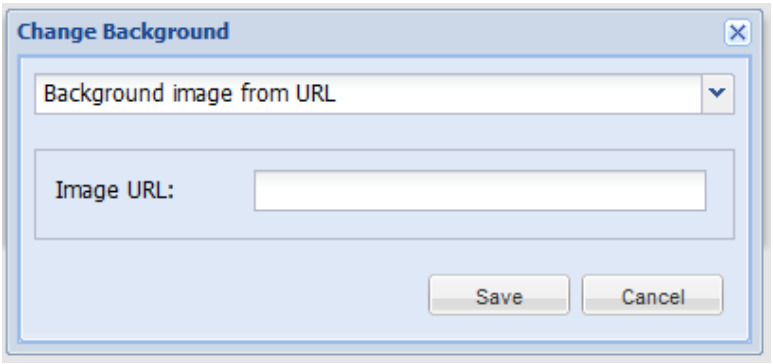
Follow these steps to create a custom view and add a background image to the view:

1. From the Datacenter View page (accessed from the Diagram selection), click **Custom View**.
2. Click **Edit** to enable edit mode.

   The Edit button highlights to indicate that it is active, and Options selections become available.
3. Select Options > Change Background.

   The Change Background dialog appears.
4. Select Background Image from URL from the list of options.
5. Enter an image location in the Image URL field, and then click **Save**. Any image format and size supported by your browser can be used.

Figure 10: Change Background



Removing the Custom View Background Image

To remove the current background image from the Custom View:

1. From the Custom View area, click Edit.
2. Select Options > Change Background.
3. In the Change Background dialog, select No background image from the list of options.
4. Click **Save**.

   The image no longer appears in the view.

Working with Devices in the Custom View

Devices in the custom view can be moved and resized. To work with devices in this view, click **Edit**. You can then drag devices to a specific location in the view, and resize them to accurately represent your physical space.

You also can view device details from this view. Click the device to go to its Status page.

Note To access device status, you cannot be in edit mode.

Removing the Custom View

Removing the custom view removes the view and custom background image, if any. To remove a custom view:

1 From the Datacenter View page (accessed from the Diagram selection), click **Custom View**.
2 Click Edit to enable edit mode.
3 Select Options > Remove Custom View.

The custom view no longer appears by default. If you select Custom View, devices still appear in the view; however, they are reset to default positions and sizes.

Activating the Auto-Generated Rack View

First, ensure that the device is included in a location. Then follow these steps to make devices visible in Datacenter View.

1 Edit the device you want to make visible. From the list of Devices, select a device (in the illustration, beta.zenoss.loc), click **Details**, and then select Edit.
2 Enter values for Rack Slot, in the format:

ru=*n*,rh=*n*,st=*n*

where:

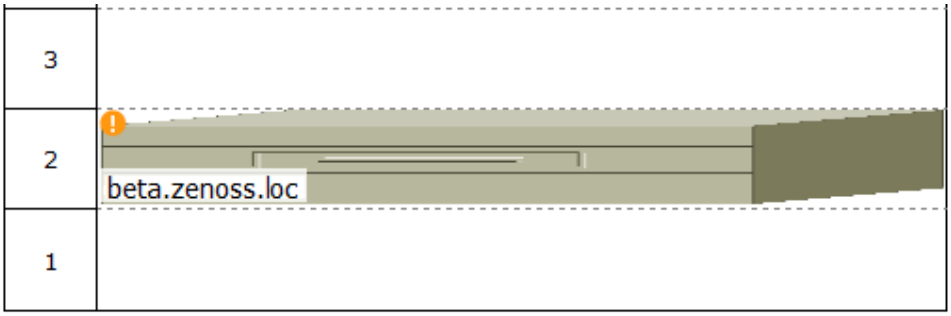
- ru=*n* sets the value for rack unit (the lowest unit used by the device)
- rh=*n* sets the value for rack height (the number of units the device uses in the rack)
- st=*n* sets the value for rack slot
- sc=*n* sets the value for slot capacity (set only for chassis devices)

For example, values of:

ru=2,rh=1

establishes a device visually in the rack as shown in this illustration:

Figure 11: Setting Rack Slot Value



Note In the example, a rack slot value is not needed, as there is only one device.

3 Click **Save**.

The device appears in Datacenter View. In the List View, it appears as part of a rack illustration. (The rack illustration is now the default image in the List View.)

In the Custom View, it appears as a single device image.

Note You can customize this device image by modifying the zIcon configuration property in the device class.

---

13 (DigMonitor) Dig Monitor

The ZenPacks.zenoss.DigMonitor ZenPack monitors the response time of DNS lookups.

To collect data, this ZenPack uses the `check_dig` Nagios plugin, which in turn uses the `dig` command.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.DigMonitor

Enable Monitoring

To enable monitoring by the system:

1. Select Infrastructure from the navigation bar.
2. Click the device name in the device list.

   The device overview page appears.
3. Expand Monitoring Templates in the left panel, and then select Device.
4. Select Bind Templates from the Action menu.

   The Bind Templates dialog appears.
5. Add the DigMonitor template to the list of selected templates, and then click **OK**.

   The DigMonitor template appears under Monitoring Templates.
6. Select the DigMonitor template in the left panel, and change options as needed.

Table 9: DigMonitor Data Source Options

Option	Description
DNS Server	Name server against which the `dig` command should be run. The default is the device host name.
Port	Port on which the name server is listening. This is normally port 53.

Option	Description
Record Name	Name of the record you want to look up. The default is zenoss.com.
Record Type	DNS record type (for example, A, MX, CNAME).

Daemons

Type	Name
Performance Collector	zencommand

14 (DistributedCollector) Distributed Collector

The ZenPacks.zenoss.DistributedCollector ZenPack allows you to deploy additional performance collection and event monitoring daemons, on the Resource Manager master host, and on other hosts.

This feature allows you to:

- Distribute processor, disk, and network load across multiple servers.
- Collect performance and events from networks that cannot be reached by the Resource Manager server.
- Configure more than one set of monitoring settings, such as different cycle times, for the `zenperfsnmp` daemon.

When you first install Distributed Collector, Resource Manager is configured with one hub and one collector. You cannot delete the initial hub and collector (each named localhost) that are set up by Distributed Collector.

About Collectors

A *collector* is a set of collection daemons, on the Resource Manager server or another server, that shares a common configuration. That configuration contains values, such as:

- Number of seconds between SNMP collections cycles
- Default discovery networks
- Maximum number of `zenprocess` parallel jobs

Each collector has its own copy of each of the Resource Manager collection daemons. For example, Resource Manager initially contains collection daemons with names like `zenperfsnmp`, `zenprocess`, and `zenping`. If you create a new collector named My2ndCollector, then the system creates new daemons named `My2ndCollector_zenperfsnmp`, `My2ndCollector_zenprocess`, and `My2ndCollector_zenping`.

About Hubs

Distributed Collector also allows you to set up new hubs. A *hub* represents an instance of the `zenhub` daemon, through which all collector daemons communicate with the object and event databases.

All collectors must belong to exactly one hub; however, a hub can have many collectors associated with it. All hubs (and indirectly all collectors) refer to the same object and event databases. Typically, only very large systems with more than five collectors (or more than 1,500 devices) benefit from multiple hubs.

Hubs manage configuration data and pass it to the collectors. Hubs also take data from the collectors and pass it to the Zenoss DataStore. Multiple hubs can be a more efficient way to manage larger deployments, as they help distribute the computing resources when configuration changes are made. They further remove the potential for configuration changes to be a bottleneck to gathering and processing data.

Typical Usage Scenarios for Distributed Monitoring

The correct distributed strategy for your environment depends on network security restrictions, as well as scale. Contact Zenoss Support if you are unsure which option best suits your enterprise.

Typical setup scenarios for using multiple hubs and collectors are shown in the following table:

Table 10: Distributed Monitoring Use Scenarios

Hub	Collector	Description
Local hub	Local collector	This setup requires only a single server, and is the most common Resource Manager deployment type. You would most likely use this configuration if you need to monitor fewer than 1000 devices, and your master Resource Manager server has direct network access to all of the monitored devices.
Local hub	Remote collector	This setup requires two servers, and is the most basic distributed setup. The primary benefit of this configuration over the local hub/local collector configuration is that the master server does no collection. This frees resources, optimizing the server's ability to perform its central role of database server and Web interface.
Local hub	Multiple remote collectors	This is the most common distributed Resource Manager configuration. You might use this configuration to scale Resource Manager to monitor more than 1000 devices. Depending on the hardware of the collectors, it is possible to monitor up to 1000 devices for each collector using this configuration. You might also use this configuration to handle differing network security policies. Often, your master Resource Manager server will not have access to all of the devices you need to monitor. In this case you can set up a remote collector with the required network access.
Multiple remote hubs	Multiple remote collectors	This configuration is for large installations only. For cases in which you have more than five collectors, you should consider deploying one or more hub servers to handle them.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.4 or higher
Required ZenPacks	ZenPacks.zenoss.DistributedCollector

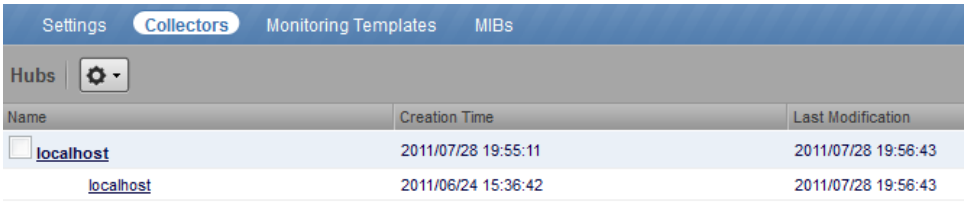
Navigating Collectors and Hubs

To view and manage collectors and hubs:

1 Log in as the Resource Manager user.
2 From the navigation menu, select Advanced > Collectors.

The Collectors page appears.

Figure 12: Collectors Page



The page lists existing hubs and collectors in hierarchical form. Hubs are listed at the top level; collectors are nested below the hub to which they belong.

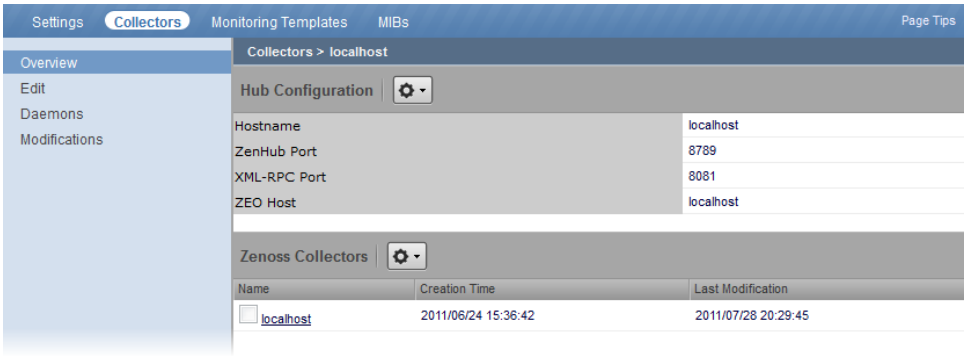
From this page, you can:

- Add a hub
- Delete a hub (which also deletes its associated collectors)
- View and edit hub settings
- Configure associated monitoring templates

Select a hub to display details and configuration options. From here, you can:

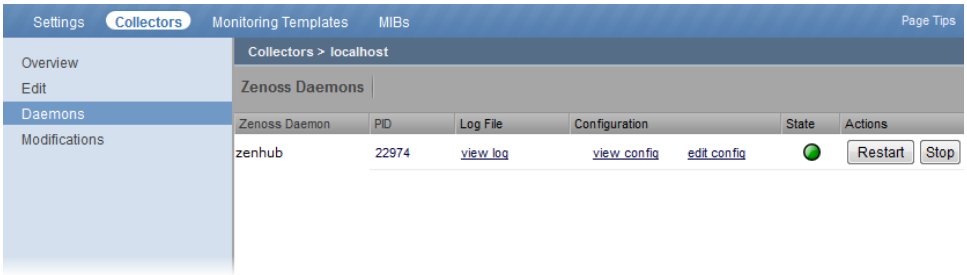
- Update a hub
- Add and delete collectors

Figure 13: Collectors Page - Overview



In the left panel, select Daemons to display details about the `zenhub` daemon that belongs to the collector. Links adjacent to the daemon name allow you to view the daemon log, and view and edit its configuration. Use the buttons to the right of the daemon name to stop, start, and restart it.

Figure 14: Collectors Page - Daemons



Updating Collectors

You must update all collectors after you:

- Update your version of Resource Manager
- Install patches
- Install, update, or remove ZenPacks
- Change the zenhub port of an associated hub

To update a collector:

1. From the navigation menu, select Advanced > Collectors.
2. Select the collector to display its Overview page.
3. Select Update Collector from the Action menu.

   Resource Manager copies the most recent code and ZenPacks to the server, and restarts the daemons running there.

Using nginx as a Reverse Proxy

Note This section is only relevant for users upgrading from Zenoss Core.

After installing Resource Manager, existing collectors must be configured to use the nginx reverse proxy. Otherwise, the host name of the collector cannot be resolved from the master, and zenwebserver (nginx) will not start.

To configure a collector to use the reverse proxy, set the render url property to:

`/remote-collector/`*CollectorID*

where *CollectorID* is the ID of the collector.

Backing Up Remote Collector Performance Data

Resource Manager does not automatically back up remote collector performance data (RRD files). To back up this data, set up a `cron` job on the remote collector. The `cron` job should invoke `zenbackup` with these options:

```
zenbackup --no-eventsdb --no-zodb
```

Old backup data is not automatically deleted; therefore, the backup solution you use to save the data should remove the backup file when it is no longer needed.

Configuring Collector Data Storage

You can configure the amount of data stored by RRDcached for each collector. Edit one or more options in the `zenrrdcached.conf` file; options are:

- **write_threads** *Value* - Specifies the number of threads for writing files. By default, this value is 4.
- **write_timeout** *Value* - Specifies the frequency at which data is written to disk. By default, this value is 300 seconds.
- **write_delay** *Value* - Specifies the delay for writing. By default, this value is 0 seconds.
- **flush_timeout** *Value* - Specifies the timeout value for flushing old data. By default, this value is 3600 seconds.

Deleting Collectors

When you delete a collector, its devices are left without an assigned collector. Zenoss recommends that you reassign assigned devices prior to deleting a collector.

To delete a collector, click the name of the hub where the collector exists from the main collectors page. The Hub overview page appears. From the list of Collectors, select the collector you want to delete. From the Action menu, select Delete Collector.

When you delete collectors using this Resource Manager instance, they are not removed or "uninstalled" in any way from the collector device. They continue to exist on the device until manually removed through the file system.

Adding Devices to Collectors

Adding devices to collectors occurs when you add the device to Resource Manager.

1. Log in to the Resource Manager user interface as a user with ZenManager or Manager privileges.
2. Navigate to **INFRASTRUCTURE > Devices**.
3. From the **Add Device** menu, select **Add a Single Device**.
4. In the **Add a Single Device** dialog, use the **Collector** field to select a local or remote collector for the device to add.
5. Populate the remaining fields as required, and then click **Add**.

The device appears in the **Devices** list, located at the bottom of the collector overview page.

Moving Devices Between Collectors

You can move devices from one collector to another.

1. Select one or more device rows in the device list.
2. Select Set Collector from the Actions list of options.
3. Select a collector, and then click **OK**.

   Resource Manager moves the devices to the selected collector.

Note If you move devices between collectors, you also can select an option choose to move their associated performance data.

Managing Collector Daemons

Collector daemons appear on the Daemons page for each collector, and can be started, stopped and restarted from there.

Only one `zentrap` 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 `zentrap` daemons. Attempting to run additional `zentrap` daemons using the same port will cause them to fail at startup.

Each collector installs the `zenrender` daemon with the rest of the collector package. If multiple collectors are installed on the same host, then the `zenrender` daemon will attempt to run for each collector. Since each `zenrender` daemon attempts to bind to the same port, all but the first daemon will fail to start. This is a problem for HA/failover environments, since failure detection systems will detect these stopped `zenrender` daemons and assume they are down. (ZEN-2967)

The remedy for this is to assign each `zenrender` daemon (beyond the first) its own unique port. This is accomplished by adding the following line to each *Collector*`_zenrender.conf` on your collector's host (in `$ZENHOME/etc`), where *Collector* is the collector name:

```
http-port 809X
```

X is a number greater than 1 and unique among multiple collectors.

Note The ports you choose for the additional collectors are arbitrary, as they are not used. However, you must leave at least one `zenrender` daemon using the default port (8091).

Specifying Daemons for Collectors

You may specify the collector daemons to start for all collectors, and for individual collectors.

- If you do not specify the daemons to start for all collectors, Resource Manager starts all collector daemons for all collectors, on the master host and on all remote collector hosts.
- If you do specify the daemons to start for all collectors, only the specified daemons are started.
- You may specify the daemons to start for all collectors, and override the global setting for individual collectors.

To specify the daemons to start for all collectors, follow these steps:

1. Log in to the Resource Manager master host as `zenoss`.
2. Open `$ZENHOME/etc/collectordaemons.txt` in a text editor.
3. List the names of collector daemons to start, one per line. The following list provides the collector daemon names. Daemons may be entered in any order.
    - zencommand
    - zeneventlog
    - zenjmx
    - zenmailtx
    - zenmodeler
    - zenperfsnmp
    - zenping
    - zenprocess
    - zenpython
    - zensyslog
    - zentrap
    - zenucsevents
    - zenvcloud
    - zenvmwareevents
    - zenvmwaremodeler
    - zenvmwareperf
    - zenvsphere
    - zenwebtx
    - zenwin
    - zenwinperf
4. Save and close the file.
5. Use the Resource Manager user interface to update each collector.

To specify the daemons to start for an individual collector, follow these steps:

1. Log in to the Resource Manager master host as `zenoss`.
2. Open `$ZENHOME/etc/collectordaemons.`*Collector-ID*`.txt` in a text editor. Substitute the ID of the collector for *Collector-ID*.
3. List the names of collector daemons to start, one per line. The preceding table provides the collector daemon names. Daemons may be listed in any order.
4. Save and close the file.
5. Use the Resource Manager user interface to update the remote collector.

SSH security information

The Distributed Collector ZenPack uses openSSH and an RSA public/private key pair for secure communications between the master host and remote hub and collector hosts.

During installation, the Distributed Collector ZenPack invokes the OpenSSH `ssh-keygen` command to generate a new, unique RSA key pair for user `zenoss` on the master host. The command generates an RSA key pair with an empty passphrase, and places the key pair in the `zenoss` user's `$HOME/.ssh` directory.

Note The Distributed Collector ZenPack does not support SSH key pairs that require a passphrase.

You may use the generated key pair to deploy a remote hub or collector, or replace the pair with a new key pair. However, you must use the same key pair for all hub and collector hosts. Therefore, if you choose to replace the key pair, Zenoss recommends doing so before deploying any remote hub or collector.

When a remote hub or collector is created, the Distributed Collector ZenPack copies the key pair of the `zenoss` user on the master host to the authorized keys file of the `zenoss` user on the remote hub or collector host, if the entry does not already exist.

When a remote hub or collector is deleted, entries for the `zenoss` user (on the master host) in the authorized keys file of the `root` and `zenoss` user on the remote hub or collector are not removed. Likewise, the known hosts file of the `zenoss` user (on the master host) is not edited to remove the entry for the remote hub or collector. You must remove these manually.

15 (DnsMonitor) DNS Monitor

The ZenPacks.zenoss.DnsMonitor ZenPack monitors the response time of DNS requests.

This ZenPack uses the `check_dns` Nagios plugin to collect data, which in turn uses the `nslookup` command.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.DNSMonitor

Enable Monitoring

To enable monitoring by the system:

1. Select Infrastructure from the navigation bar.
2. Click the device name in the device list.

   The device overview page appears.
3. Expand Monitoring Templates in the left panel, and then select Device.
4. Select Bind Templates from the Action menu.

   The Bind Templates dialog appears.
5. Add the DNSMonitor template to the list of selected templates, and then click **OK**.

   The DNSMonitor template appears under Monitoring Templates.
6. Select the DNSMonitor template in the left panel, and change options as needed.

Table 11: DNSMonitor Data Source Options

Option	Description
DNS Server	Name server against which the `nslookup` command should be run. If empty (the default), the default DNS server or servers in `/etc/resolve.conf` are used.
Port	Port on which the name server is listening. This is normally port 53.

Option	Description
Host Name	Host name to resolve. The default is the device ID.
Expected IP Address	IP address to which the host name is expected to resolve.

Daemons

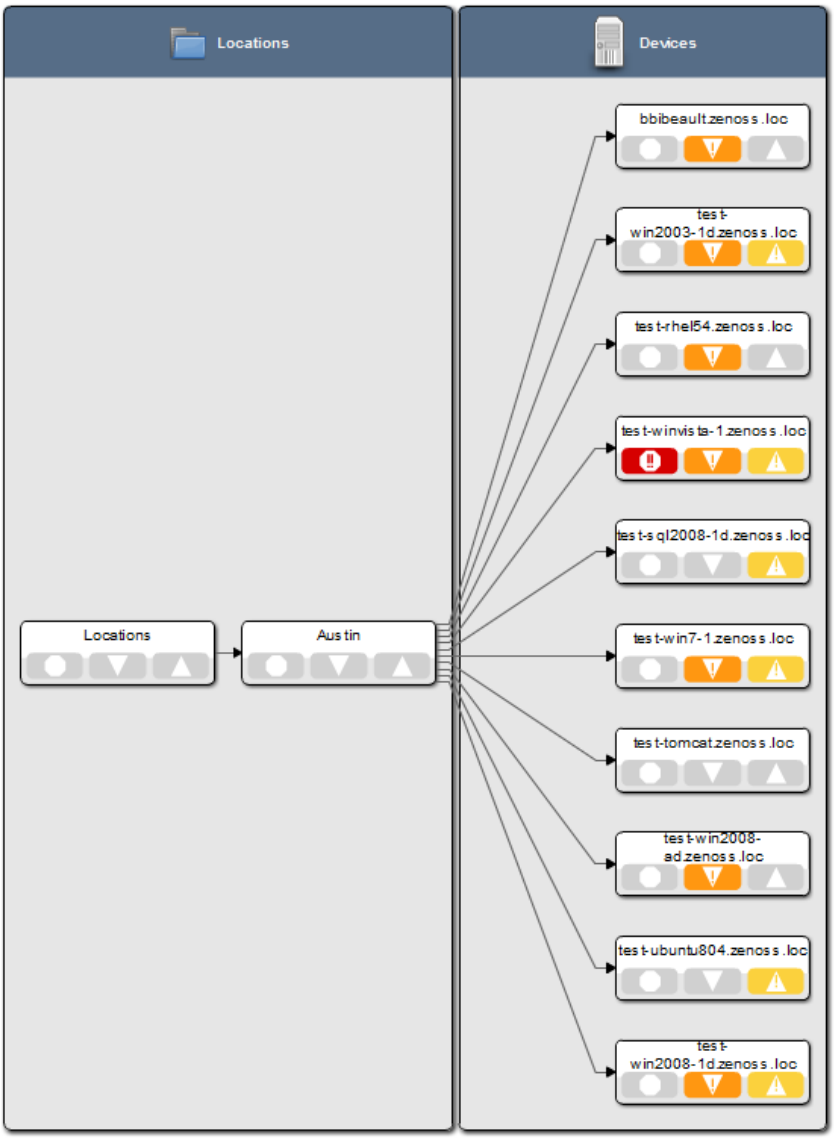
Type	Name
Performance Collector	zencommand

16 (DynamicView) Dynamic Service View

The ZenPacks.zenoss.DynamicView ZenPack provides a dynamic visualization of system objects and their relationships to other objects.

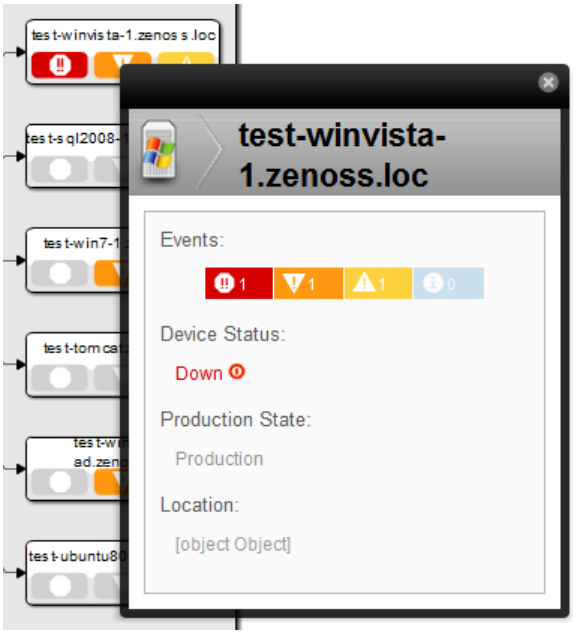
You can access the dynamic view from groups, systems, and locations. Depending on the object type, different relationships are illustrated. Each dynamic view shows related objects in a graph. Each object in that graph displays its associated event information.

Figure 15: Dynamic Service View: Locations Graph



When you click an object in the graph, the "inspector" panel appears. This panel provides detailed information about the object and links directly to it. Information that appears in the inspector depends on the object type selected.

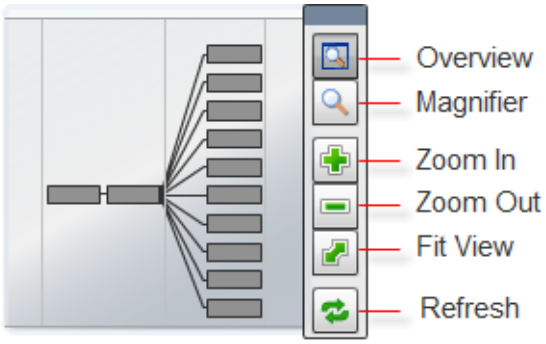
Figure 16: Dynamic Service View: Inspector Panel



View controls appear to the right of the graph. These allow you to adjust your view:

- **Overview** - Toggles display on and off of the graph overview illustration.
- **Magnifier** - Toggles on and off the magnifier, which allows you to magnify selected portions of the graph.
- **Zoom In** - Zooms in on the graph.
- **Zoom Out** - Zooms out on the graph.
- **Fit View** - Fits the graph to the browser page.
- **Refresh** - Refreshes the graph.

Figure 17: Dynamic View: View Controls



Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 3.0 or higher
Required ZenPacks	ZenPacks.zenoss.DynamicView
Other	Oracle JRE 1.5 or later, Flash-enabled Web browser

Dynamic View of Organizers

The dynamic view of organizers shows objects that can impact the status of the organizer, such as other organizers and devices. This view also shows relationships between devices and a virtual infrastructure, such as VMware or Cisco UCS objects monitored by the system, as well as storage information.

To access the dynamic view for an organizer (such as a group, system, or location):

1 From Infrastructure > Devices, select the organizer in the devices hierarchy.
2 Click **Details**.
3 Select Dynamic Service View.

Dynamic View of Devices

The dynamic view of devices shows the relationship between a device and monitored components.

To access the dynamic view for a device:

1 From Infrastructure > Devices, click a device in the device list.

   The device overview page appears.
2 Select Dynamic Service View in the left panel.

Dynamic View of Cisco UCS Devices

On Cisco UCS devices, the dynamic view shows the components and relationships that make up a Cisco UCS cluster.

Dynamic View of VMware Hosts

On VMware Hosts (ESX servers), the dynamic view shows the relative VMware elements that are connected to the host, such as:

- VMs that currently are running on the Host
- Data stores that are mounted by the Host
- Clusters to which the Host belongs

Dynamic View of Storage Devices

On storage devices, such as NetApp Filers, there are two dynamic views:

- **Physical Storage View** - Shows the device's storage enclosures and associated hard disks.
- **Logical Storage View** - Shows the logical storage arrangement that the storage device presents, such as file systems and raid groups.

Daemons

This ZenPack adds the `zenjserver` daemon to Resource Manager.

Type	Name
Display	`zenjserver`

17 (EnterpriseCollector) Enterprise Collector

The ZenPacks.zenoss.EnterpriseCollector ZenPack allows several collector daemons to start and to monitor devices, even if a connection to `zenhub` is not available when a collector daemon starts.

With this ZenPack, the following collector daemons gain configuration caching:

- `zenwin`
- `zeneventlog`
- `zenwinperf`
- `zenprocess`

Data and events are cached locally and are sent to `zenhub` as needed after a connection is re-established. Cached configuration data is stored in `$ZENHOME/perf/Daemons/MonitorName/DaemonName-`*Suffix*, where *Suffix* is one of:

- `configs.db`
- `properties.pickle`
- `threshold-classes.pickle`
- `thresholds.pickle`

For example:

```
[zenoss@zenosst zenpacks]$ ls $ZENHOME/perf/Daemons/localhost/
zeneventlog*
/opt/zenoss/perf/Daemons/localhost/zeneventlog-configs.db
/opt/zenoss/perf/Daemons/localhost/zeneventlog-properties.pickle
/opt/zenoss/perf/Daemons/localhost/zeneventlog-threshold-classes.pickle
/opt/zenoss/perf/Daemons/localhost/zeneventlog-thresholds.pickle
```

Each time a collector daemon successfully retrieves configuration information from `zenhub`, it updates the cached files. This happens at startup, and then every 20 minutes to 6 hours (depending on the daemon and its configuration). A collector daemon must successfully connect once before it can use the cached files if `zenhub` is not available.

The cached files are considered transient, and can be deleted without harm.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.5 or higher
Required ZenPacks	ZenPacks.zenoss.EnterpriseCollector

Configuration Options

These options apply to all collector daemons and control how those daemons request configurations from `zenhub` daemons:

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

  By requesting device configurations in batches, the option allows the collector daemon to start monitoring devices as the device configuration is loaded. Also, the smaller batches prevent `zenhub` daemons 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
```

18 (EnterpriseLinux) Enterprise Linux

The ZenPacks.zenoss.EnterpriseLinux ZenPack extends the capabilities of ZenPacks.zenoss.LinuxMonitor and enables Resource Manager to use Secure Shell (SSH) to monitor Linux hosts.

Resource Manager models and monitors devices placed in the `/Server/SSH/Linux` device class by running commands and parsing the output. Parsing of command output is performed on the Resource Manager server or on a distributed collector. The account used to monitor the device does not require root access or special privileges for the default modeler plugins.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.4 or higher
Required ZenPacks	ZenPacks.zenoss.LinuxMonitor, ZenPacks.zenoss.EnterpriseLinux

Add a Linux Server

The following procedure assumes that the credentials have been set.

1 From Infrastructure > Devices, Select Add a Single Device from the Add Device menu.
2 Enter the following information:

Table 12: Adding Linux Device Information

Name	Description
Device Name	Linux host to model
Device Class Path	`/Server/SSH/Linux`
Discovery Protocol	Set this to `auto` unless adding a device with username/password different than found in the device class. If you set this to `none`, then you will need to add the credentials, and then manually model the device.

3 Click **Add**.

Related Links

Set Linux Server Monitoring Credentials on page 78

Set Linux Server Monitoring Credentials

All Linux servers must have a device entry in an organizer below the `/Devices/Server/SSH/Linux` device class.

Note The SSH monitoring feature will attempt to use key-based authentication before using a configuration properties password value.

1. Navigate to the device or device class in the Resource Manager interface.
   - If applying changes to a device class:
     1. Select the class in the devices hierarchy.
     2. Click **Details**.
     3. Select Configuration Properties.
   - If applying changes to a device:
     1. Click the device in the device list.
     2. Select Configuration Properties.
2. Verify the credentials for the service account to access the service.

Table 13: Linux Configuration Properties

Name	Description
zCommandUsername	Linux user with privileges to gather performance information.
zCommandPassword	Password for the above user.

3. Click **Save** to save your changes.

Resolving CHANNEL_OPEN_FAILURE Issues

The `zencommand` daemon's log file (`$ZENHOME/collector/zencommand.log`) may show messages stating:

```
ERROR zen.SshClient CHANNEL_OPEN_FAILURE: Authentication failure
WARNING:zen.SshClient:Open of command failed (error code 1): open failed
```

If the `sshd` daemon's log file on the remote device is examined, it may report that the `MAX_SESSIONS` number of connections has been exceeded and that it is denying the connection request. At least in the OpenSSH daemons, this `MAX_SESSIONS` number is a compile-time option and cannot be reset in a configuration file.

In order to work around this limitation of the `sshd` daemon, use the configuration property *zSshConcurrentSessions* to control the number of connections created by `zencommand` to the remote device.

1. Navigate to the device or device class in the Resource Manager interface.
   - If applying changes to a device class:
     1. Select the class in the devices hierarchy.

2. Click **Details**.
3. Select Configuration Properties.

- If applying changes to a device:

1. Click the device in the device list.
2. Select Configuration Properties.

2 Apply an appropriate value for the maximum number of sessions.

Table 14: Concurrent SSH Configuration Properties

Name	Description
zSshConcurrentSessions	Maximum number of sessions supported by the remote device's `MAX_SESSIONS` parameter. A common value for Linux is 10.

3 Click **Save** to save your changes.

Resolving Command timed out Issues

The `zencommand` daemon's log file (`$ZENHOME/collector/zencommand.log`) may show messages stating:

```
WARNING:zen.zencommand:Command timed out on device device_name: command
```

If this occurs, it usually indicates that the remote device has taken too long to return results from the commands. To increase the amount of time to return results, change the configuration property *zCommandCommandTimeout* to a larger value.

1 Navigate to the device or device class in the Resource Manager interface.

- If applying changes to a device class:

1. Select the class in the devices hierarchy.
2. Click **Details**.
3. Select Configuration Properties.

- If applying changes to a device:

1. Click the device in the device list.
2. Select Configuration Properties.

2 Apply an appropriate value for the command timeout.

Table 15: SSH Timeout Configuration Properties

Name	Description
zCommandCommandTimeout	Time in seconds to wait for commands to complete on the remote device.

3 Click **Save** to save your changes.

DMIDECODE Modeler Plugin

This plugin allows you to collect and model detailed hardware and kernel information on your Linux devices.

Since the `dmidecode` command requires root privileges, it needs to be run with something like `sudo`. Sample entries required on the sudoers file on each remote device are:

```
Cmnd_Alias DMIDECODE = /usr/sbin/dmidecode
## Allows members of the zenoss group to gather modeling information
Defaults:zenoss !requiretty
%zenoss ALL = (ALL) NOPASSWD: DMIDECODE
```

To use this plugin, add it to the list of collector plugins for the device or device class, and then remodel. For more information about working with Resource Manager plugins, refer to *Zenoss Service Dynamics Resource Manager Administration*.

Daemons

Type	Name
Modeler	zenmodeler
Performance Collector	zencommand

19 (EnterpriseReports) Enterprise Reports

The ZenPacks.zenoss.EnterpriseReports ZenPack adds a variety of reports to Resource Manager.

Reports include:

- Organizer Availability
- Organizer Graphs
- 95th Percentile
- Users Group Membership
- Maintenance Windows
- Notifications and Triggers by Recipient
- Interface Volume
- Event Time to Resolution
- User Event Activity
- Datapoints per Collector
- Interface Utilization
- Data Sources in Use
- Defined Thresholds
- Network Topology
- Customized Performance Templates
- Cisco Inventory
- Guest to Datapools

To access the reports, select Reports from the Navigation bar. The reports appear in the left panel.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.EnterpriseReports

Organizer Availability

Provides the availability percentage of all network organizers in the system. This report can be filtered by organizer, event class, component, and date.

You can report on the availability of device classes, locations, systems, or groups within a defined time frame. This report offers two reporting modes:

- **Averaged** - Defines the organizer as available for the average availability time for all devices contained in it.
- **Coalesced** - Defines the organizer as available only if all devices are available during a certain time period.

Two modes of operation: Averaged - defines the organizer as available for the average availability time for all the devices contained within it. Coalesced - defines availability of the organizer as the available only if all devices are available during a certain time period.

95th Percentile

The 95th Percentile report provides details about all network interfaces in the system, sorted by highest utilization.

95th percentile is a widely used mathematical calculation that evaluates the regular and sustained utilization of a network connection. The 95th percentile method more closely reflects the needed capacity of the link in question than other methods (such as mean or maximum rate).

This report is useful for network capacity planning and billing for either average or 95th percentile bandwidth utilization.

You can filter this report by device name. Enter a complete or partial name (using * (asterisk) for matching), and then click **Update** to filter the report.

To change the reporting time period, enter Start and End dates (or click **Select** to select dates from a calendar). Click **Update** to refresh the report.

Users Group Membership

Shows all users and the groups to which they belong.

Maintenance Windows

The Maintenance Windows report shows all defined windows that are active during a selected time period.

To change the reporting time period, enter Start and End dates (or click **Select** to select dates from a calendar). Click **Update** to refresh the report.

Interface Volume

The Interface Volume report shows network interface volume. It reports on all network interfaces in the system, sorted by highest utilization. Volume is defined as the total number of bytes transferred during a specific reporting period.

This report is useful for determining billing on total bandwidth consumption.

To change the reporting time period, enter Start and End dates (or click **Select** to select dates from a calendar). Click **Update** to refresh the report.

Event Time to Resolution

The Event Time to Resolution report shows, for each user, the total time taken to acknowledge or clear events. Results are organized by event severity.

This report is helpful for tracking response time SLAs in a NOC-type environment.

User Event Activity

Reports the total number of events acknowledged and cleared, on a per-user basis, during the reporting period.

This report is helpful for tracking operator activity in a NOC-type environment.

Datapoints Per Collector

Shows the number of devices and data points per collector, which is useful for gauging how much monitoring load is on each collector.

Defined Thresholds

The Defined Thresholds report provides details about all thresholds defined in the system. The report links to the target of each threshold. The target can be a device class, individual device, or individual component.

This report is useful for administering the system. You can use it to quickly identify which threshold events can occur within the system, and the severity of those events.

Network Topology

Shows the layout of the network, according to the routes that Resource Manager understands, starting from the collector and ending at the remote devices associated with the collector.

The report does not return data if the host on which the Resource Manager collector is running does not have a device created in the DMD. Create a device representing the collector in the DMD, and then run report again.

An invalid route entry (for example, 'Missing link here' value in the Route column) indicates that Resource Manager cannot determine how to route from one device to another. Correct this by adding a network interface to the model (no new hardware required) and then adding a new route entry from the last device in the route to the device (the IP address shown at the far right of the table).

(EnterpriseSecurity) Enterprise Security 20

The ZenPacks.zenoss.EnterpriseSecurity ZenPack enhances Resource Manager security by encrypting stored passwords.

Resource Manager stores the passwords it uses to remotely access hosts in a Zope Object Database (ZODB). After enabling this feature, these passwords are encrypted according to the Advanced Encryption Standard (AES), with 256-bit key sizes.

By using the password encryption feature, you can help prevent an attacker from accessing your managed systems if he gains access to a backup copy of your ZODB.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.5 or higher
Required ZenPacks	ZenPacks.zenoss.EnterpriseSecurity

Testing Password Encryption

To test password encryption, use the `grep` command to search the `Data.fs` file for the value of one of the password configuration properties. For example, if you set zCommandPassword to a value of wobet51, you can check that passwords are encrypted by using this command on the Resource Manager master host:

```
strings $ZENHOME/var/Data.fs | grep wobet51
```

If password encruption is working correctly, this command returns no results.

21 (EsxTop) VMware ESX Server

The ZenPacks.zenoss.EsxTop ZenPack uses the VMware `resxtop` command to gather performance information about VMware Infrastructure™ ESX™ servers.

Note This ZenPack is deprecated; see *(vSphere) VMware vSphere* on page 177.

This ZenPack can be used alone, or with one of the other VMware ZenPacks. When used alone, a basic modeler creates virtual machines under the `/Devices/Server/Virtual Hosts/EsxTop` device class for any host device that is added and modeled. Otherwise, performance data can be collected for the ESX hosts modeled by the other ZenPacks.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x
Required ZenPacks	ZenPacks.zenoss.ZenossVirtualHostMonitor v2.3.5 ZenPacks.zenoss.EsxTop
Required Software (on collectors)	OpenSSL development package v0.9.7 VMware vSphere CLI v4.1

Installing Prerequisite Libraries

The VMware vSphere CLI is required for access to the `resxtop` command, which enables Resource Manager to model and gather performance information about individual ESX servers.

Follow these steps to install the CLI and required software:

1. If you have not yet installed it, install the OpenSSL development package. For example, for an RPM-based system, enter:

```
yum install openssl-devel
```

2. From your VMware account, download the VMware vSphere CLI.

Note For downloads and documentation, go to:

http://downloads.vmware.com/d/details/vcli41/ZHcqYmRoaCpiZHRAag==

3 Copy the package to each Resource Manager collector.
4 For each collector:

a Expand the package file.
b Run the following command to install the package:

```
./vmware-install.pl
```

c As the zenoss user, run the following command to verify successful installation:

```
resxtop --server myESXServer --user
 userOnRemoteEsxServerAllowedToUseEsxTop -b -n 1 -a
```

The `resxtop` command prompts for a password.
d Enter the password for a user with permissions on the remote ESX server.

If the command is working correctly, then a screen displays with several pages of command output.
e Create a symbolic link from the location that the `resxtop` command was installed into the `$ZENHOME/libexec` directory. This allows the `check_esxtop` command to automatically determine which binary to run. For example:

```
cd $ZENHOME/libexec
ln -s PathToResxtop
```

f Test the `check_esxtop` command by showing the VMs on the remote server:

```
$ZENHOME/ZenPacks/Ze*EsxTop*/Z*/z*/E*/libexec/check_esxtop --
server=myEsxserver \
 --user=userOnRemoteEsxServerAllowedToUseEsxTop --password=password
 --showvms
```

Enabling the ZenPack

Follow these steps to enable this ZenPack. From the Resource Manager interface, add a host:

1 From Infrastructure > Devices, navigate to the `/Devices/Server/Virtual Hosts/EsxTop` device class.
2 From the Add Device menu, select Add a Single Device.

The Add a Single Device dialog appears.
3 Enter a host name or IP address.
4 De-select the Model Device option.
5 Click **Add**.
6 Select the newly added device in the list.

The device overview appears.
7 Click **Details**, and then select Configuration Properties in the left panel.
8 Enter login credentials for the zCommandUsername and zCommandPassword configuration properties, and then click **Save**.

9 If the device has an SNMP agent installed, update the ESX device configuration with the appropriate SNMP configuration information, and then add any desired modeler plugins.

10 From the Action menu, select Model device.

Daemons

Type	Name
Modeler	`zenmodeler`
Performance Collector	`zencommand`

(FoundryMonitor) Foundry Networks Devices 22

The ZenPacks.zenoss.FoundryMonitor ZenPack monitors networking devices built by Foundry Networks (now Brocade Communication Systems).

Note This ZenPack is not installed when Resource Manager is installed. To download it, visit the *Zenoss Support* site.

This ZenPack models specific details on Foundry devices, including:

- DRAM
- Serial Number
- Processor
- Product type

This ZenPack monitors memory utilization, as well as CPU utilization averages for 1 minute, 1 second, and 5 seconds.

It also includes all Foundry traps to ensure proper decoding of those traps through `zentrap`.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x
Required ZenPacks	ZenPacks.zenoss.FoundryMonitor

Configuring Resource Manager

All Foundry devices must exist in the `/Devices/Network/Foundry` device class.

Follow these steps to configure Resource Manager:

1 Navigate to the device or device class in the Resource Manager interface.

- If applying changes to a device class:

  1 Select the class in the devices hierarchy.
  2 Click **Details**.
  3 Select Configuration Properties.

- If applying changes to a device:

  1. Click the device in the device list.
  2. Select Configuration Properties.

2. Edit the appropriate configuration properties for the device or devices.

Table 16: Foundry Configuration Properties

Name	Description
zSnmpCommunity	Consult with your network administrators to determine the SNMP community permitted.
zSnmpMonitorIgnore	Set to a value of `False`.
zSnmpPort	The default port is 161.
zSnmpVer	Set to a value of `v2c`.

3. Click **Save** to save your changes. Resource Manager now will begin collecting Foundry device metrics from this device.
4. Navigate to Graphs and you should see some placeholders for performance graphs. After approximately fifteen minutes you should see the graphs start to become populated with information.

Daemons

Type	Name
Modeler	`zenmodeler`
Performance Collector	`zenperfsnmp`
Traps	`zentrap`

(FtpMonitor) FTP Monitor 23

The ZenPacks.zenoss.FtpMonitor ZenPack monitors the response times of File Transfer Protocol (FTP) server connection requests.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.FtpMonitor

Enable Monitoring

To enable monitoring of the device:

1. Select Infrastructure from the navigation bar.
2. Click the device name in the device list.

   The device overview page appears.
3. Expand Monitoring Templates in the left panel, and then select Device.
4. Select Bind Templates from the Action menu.

   The Bind Templates dialog appears.
5. Select the FTPMonitor template and move it to the list of selected templates.
6. Click **Save**.

   The FTPMonitor template appears under Monitoring Templates.
7. Select the FTPMonitor template and change options as needed.

Table 17: FTPMonitor Basic Data Source Options

Option	Description
Port	The port to connect to FTP server (default 21)
Send String	Command string to send to the server

Option	Description
Expect String	A string to expect in server response
Mismatch	If the expected string does not match the string returned from the remote server, create an event with one of these states: ok, warn, crit (default: warn)
Quit String	Command to send to the remote server to end the session

Enable Secure Site Monitoring

To enable secure site monitoring:

1. Select Infrastructure from the navigation bar.
2. Click the device name in the devices list.

   The device overview page appears.
3. Expand Monitoring Templates in the left panel.
4. Select the FTPMonitor template and change options as needed.

Table 18: FTPMonitor Secure Data Source Options

Option	Description
Port	The port to connect to FTP server (default 21).
Certificate	Minimum days for which a certificate is valid
Use SSL	Use SSL for the connection

Tuning for Site Responsiveness

1. Select Infrastructure from the navigation bar.
2. Click the device name in the devices list.

   The device overview page appears.
3. Expand Monitoring Templates in the left panel.
4. Select the FTPMonitor template and change options as needed.

Table 19: FTPMonitor Tunables Data Source Options

Option	Description
Timeout	Seconds before connection times out (default: 60)
Refuse	If a TCP/IP connection to the remote service is refused (ie no program is listening at that port) send an event with one of these severity states: ok, warn, crit (default: crit)
Max Bytes	Close the connection once more than this number of bytes are received.
Delay	Seconds to wait between sending string and polling for response

Option	Description
Warning response time (seconds)	Response time to result in a warning status.
Critical response time (seconds)	Response time to result in critical status

Daemons

Type	Name
Performance Collector	zencommand

(HPMonitor) HP Monitor 24

The ZenPacks.zenoss.HPMonitor ZenPack provides customized modeling of devices running HP Insight Management Agents, and includes identification for proprietary HP hardware.

The following information is collected through SNMP:

- Hardware Model
- Hardware Serial Number
- Operating System
- CPU Information (socket, speed, cache)

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.HPMonitor
On each remote device	The HP Insight SNMP Management Agent gathers information about the device.

Enable Enhanced Modeling

To enable enhanced modeling:

1. Select Infrastructure from the navigation bar.
2. Click the device name in the device list.

   The device overview page appears.
3. Select Modeler Plugins from the left panel.
4. Click Add Fields to reveal the list of available plugins.
5. Select the following available plugins and drag them to the plugins list:

   - HPCpuMap
   - HPDeviceMap
6. Remove the following plugins by clicking the 'X' button to the right of the plugin:

- zenoss.snmp.CPUMap
- zenoss.snmp.DeviceMap

7 Click **Save**.

8 Remodel the device using the new plugins. To do this, select Model Device from the Action menu.

Daemons

Type	Name
Modeler	zenmodeler
Performance Collector	zenperfsnmp

25 (HpuxMonitor) HP-UX Monitor

The ZenPacks.zenoss.HpuxMonitor ZenPack monitors HP UNIX (HP-UX) servers through Secure Shell (SSH).

Resource Manager models and monitors devices placed in the `/Server/SSH/HP-UX` device class by running commands and parsing the output. Command output is parsed on Resource Manager collectors. The account used to monitor the device requires root access or special privileges to access `/usr/bin/adb`.

This ZenPack provides:

- File system and process monitoring
- Network interfaces and route modeling
- CPU utilization information
- Hardware information (memory, number of CPUs, and model numbers)
- OS information (OS-level, command-style information)
- Software package information (such as installed software)

Prerequisites

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

Note If using a distributed collector setup, SSH requires firewall access (by default, port 22) from the collector to the monitored server.

Limitations

This ZenPack has not been tested on Itanium systems.

Add an HP-UX Device for Monitoring

These steps assume that credentials have been set.

1 From Infrastructure > Devices, select Add a Single Device from the Add Device menu.
2 Enter the following information:

Table 20: Adding HP-UX Device Information

Name	Description
Name or IP	HP-UX host to model
Device Class	`/Server/SSH/HP-UX`
Model Device	Select this option unless adding a device with a user name and password different than found in the device class. If you de-select this option, then you must add the credentials, and then manually model the device.

3 Click **Add Device** to add the device.

Related Links

Set HP-UX Server Monitoring Credentials on page 96

Set HP-UX Server Monitoring Credentials

All HP-UX servers must have a device entry in an organizer below the `/Devices/Server/SSH/HP-UX` device class.

Note The SSH monitoring feature will attempt to use key-based authentication before using a configuration properties password value.

Set Credentials for the Device

1 In the Web interface, navigate to the device.
2 In the left panel, select Configuration Properties.
3 Verify the credentials for the service account to access the service:

Table 21: HP-UX Configuration Properties

Name	Description
zCommandUsername	HP-UX user with privileges to gather performance information
zCommandPassword	Password for the HP-UX user

4 Click **Save** to save your changes.

Set Credentials for the Device Class

1 In the Web interface, navigate to the `Devices/Server/SSH/HP-UX` device class.
2 In the left panel, select Configuration Properties.
3 Verify the credentials for the service account to access the service. (Refer to the previous table titled "HP-UX Configuration Properties.")
4 Click **Save** to save your changes.

Resolving CHANNEL_OPEN_FAILURE Issues

The zencommand daemon's log file ($ZENHOME/*collector*/zencommand.log) may show messages stating:

```
ERROR zen.SshClient CHANNEL_OPEN_FAILURE: Authentication failure
WARNING:zen.SshClient:Open of command failed (error code 1): open failed
```

If you view the sshd daemon's log file on the remote device, you may see that the MAX_SESSIONS number of connections has been exceeded and that it is denying the connection request. In the OpenSSH daemons, this MAX_SESSIONS number is a compile-time option and cannot be reset in a configuration file.

To work around this sshd daemon limitation, use the configuration property *zSshConcurrentSessions* to control the number of connections created by zencommand to the remote device:

1 Navigate to the device or device class in the Resource Manager interface.

- If applying changes to a device class:
  1 Select the class in the devices hierarchy.
  2 Click **Details**.
  3 Select Configuration Properties.
- If applying changes to a device:
  1 Click the device in the device list.
  2 Select Configuration Properties.

2 Apply an appropriate value for the maximum number of sessions.

Table 22: Concurrent SSH Configuration Properties

Name	Description
zSshConcurrentSessions	Maximum number of sessions supported by the remote device's MAX_SESSIONS parameter. Common values for HP-UX are 2 and 10.

3 Click **Save** to save your changes.

Resolving Command time out Issues

The zencommand daemon's log file ($ZENHOME/*collector*/zencommand.log) may show messages stating:

```
WARNING: zen.zencommand:Command timed out on device device_name: command
```

If this occurs, it generally indicates that the remote device has taken too long to return results from the commands. To increase the amount of time to allow devices to return results, change the configuration property *zCommandCommandTimeout* to a larger value:

1 Navigate to the device or device class in the Resource Manager interface.

- If applying changes to a device class:
  1 Select the class in the devices hierarchy.
  2 Click **Details**.

3. Select Configuration Properties.

- If applying changes to a device:

1. Click the device in the device list.
2. Select Configuration Properties.

2. Apply an appropriate value for the command timeout.

Table 23: SSH Timeout Configuration Properties

Name	Description
zCommandCommandTimeout	Time in seconds to wait for commands to complete on the remote device.

3. Click **Save** to save your changes.

Daemons

Type	Name
Modeler	zenmodeler
Performance Collector	zencommand

(HttpMonitor) HTTP Monitor 26

The ZenPacks.zenoss.HttpMonitor ZenPack monitors the response times of HTTP server connection requests, and determines whether specific content exists on a Web page.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.HttpMonitor

Enable Monitoring

Follow these steps to enable monitoring:

1. Select Infrastructure from the navigation bar.
2. Click the device name in the device list.

   The device overview page appears.
3. Expand Monitoring Templates, and then select Device from the left panel.
4. Select Bind Templates from the Action menu.

   The Bind Templates dialog appears.
5. Add the HttpMonitor template to the list of selected templates, and then click **Submit**.

> **Note** Prior to Zenoss 2.4, this template was not available. If your version is prior to Zenoss 2.4 you must create the template, data source and graphs manually. For more information, refer to *Zenoss Service Dynamics Resource Manager Administration*.

The HttpMonitor template is added to the list of monitoring templates. You can now begin collecting Web server metrics from the device.

Check for a Specific URL or Specify Security Settings

1. Select Infrastructure from the navigation bar.
2. Click the device name in the device list.

The device overview page appears.

3 Expand Monitoring Templates, and then select Device from the left panel.
4 Create a local copy of the template.
5 Select the newly created local template copy.
6 Select the HttpMonitor data source, and then select View and Edit Details from the Action menu.

The Edit Data Source dialog appears.

7 Change data source options as needed, and then click **Save**.

Table 24: HTTPMonitor Content Checking Data Source Options

Option	Description
Port	The port to connect to HTTP server (default 80).
Use SSL	Use SSL for the connection
Url	Address of the web page.
Basic Auth User	If the website requires credentials, specify the username here.
Basic Auth Password	Password for the user.
Redirect Behavior	If the web site returns an HTTP redirect, should the probe follow the redirect or create an event? Possible event severities are `OK`, `Warning`, and `Critical`.

Check for Specific Content on the Web Page

This procedure allows Resource Manager to create an event if content at the web page does not match the expected output.

1 Select Infrastructure from the navigation bar.
2 Click the device name in the device list.

The device overview page appears.

3 Expand Monitoring Templates, and then select Device from the left panel.
4 Create a local copy of the template.
5 Select the newly created local template copy.
6 Select the HttpMonitor data source, and then select View and Edit Details from the Action menu.

The Edit Data Source dialog appears.

7 Change data source options as needed, and then click **Save**.

Table 25: HTTPMonitor Content Checking Data Source Options

Option	Description
Regular Expression	A Python regular expression to match text in the web page.
Case Sensitive	Is the regular expression case-sensitive or not?

Option	Description
Invert Expression	If you would like to test to see if the web page does **not** contain content matched by a regular expression, check this box.

Tuning for Site Responsiveness

1. Select Infrastructure from the navigation bar.
2. Click the device name in the device list.

   The device overview page appears.
3. Expand Monitoring Templates, and then select Device from the left panel.
4. Create a local copy of the template.
5. Select the newly created local template copy.
6. Select the HttpMonitor data source, and then select View and Edit Details from the Action menu.

   The Edit Data Source dialog appears.
7. Change data source options as needed, and then click **Save**.

Table 26: HTTPMonitor Tunables Data Source Options

Option	Description
Timeout (seconds)	Seconds before connection times out (default: 60)
Cycle Time (seconds)	Number of seconds between collection cycles (default: 300 or five minutes)

Daemons

Type	Name
Performance Collector	zencommand

(IISMonitor) Microsoft Internet Information Server 27

The ZenPacks.zenoss.IISMonitor ZenPack uses Windows Perfmon to collect performance metrics from Microsoft Internet Information Server (IIS).

Note This ZenPack is deprecated; see *(Microsoft.Windows) Microsoft Windows* on page 123.

No agent need be installed on IIS servers to collect the following metrics.

- Connections Attempts
- Throughput (Bytes & Files)
- Requests (GET, HEAD, POST, CGI, ISAPI)
  - Standard: GET, HEAD, POST, CGI, ISAPI
  - WebDAV: PUT, COPY, MOVE, DELETE, OPTIONS, PROPFIND, PROPPATCH, MKCOL
  - Other: SEARCH, TRACE, LOCK, UNLOCK

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x
Required ZenPacks	ZenPacks.zenoss.WindowsMonitor, ZenPacks.zenoss.IISMonitor

Enable Monitoring

All IIS servers must have a device entry in an organizer below the `/Devices/Server/Windows/WMI` device class. In addition, verify that your Resource Manager Windows service account has access to the IIS service.

1 Bind the IIS template to the `/Devices/Server/Windows/WMI` class. To do this:

   a Select the device class in the devices hierarchy.
   b Click **Details**.
   c Select Device_WMI under Monitoring Templates.
   d Select Bind Templates from the Action menu.

The Bind Templates dialog appears.

e Move IIS (/Server/Windows/WMI) from the Available area to the Selected area, and then click **Save**.

2 Navigate to the device or device class in the Resource Manager interface.

- If applying changes to a device class:
  1 Select the class in the devices hierarchy.
  2 Click **Details**.
  3 Select Configuration Properties.
- If applying changes to a device:
  1 Click the device in the device list.
  2 Select Configuration Properties.

3 Verify the credentials for the service account to access the service.

Table 27: IIS Configuration Properties

Name	Description
zWinUser	Windows user with privileges to gather performance information.
zWinPassword	Password for the above user.

4 Click **Save** to save your changes.

You will now be able to start collecting the IIS server metrics from this device.

5 Navigate to Graphs and you should see some placeholders for graphs. After approximately fifteen minutes you should see the graphs begin to be populated with information.

Note For more information about user credentials and troubleshooting WMI connections, see *WindowsMonitor (Microsoft Windows)* on page 197.

Daemons

Table 28: Daemons

Type	Name
Performance Collector	`zenwinperf`

(IRCDMonitor) IRCD Monitor 28

The ZenPacks.zenoss.IRCDMonitor ZenPack monitors the number of users connected to an Internet Relay Chat (IRC) server.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.IRCDMonitor

Enable Monitoring

To enable monitoring:

1. Select Infrastructure from the navigation bar.
2. Click the device name in the device list.

   The device overview page appears.
3. Expand Monitoring Templates in the left panel, and then select Device.
4. Select Bind Templates from the Action menu.

   The Bind Templates dialog appears.
5. Move the IrcdMonitor template from the Available list and move it to the Selected list.
6. Click **Save**.

   The IrcdMonitor template is added.
7. Click the new template in the left panel and change options as needed.

Table 29: IRC Basic Data Source Options

Option	Description
Port	Specifies the port to connect to the IRC server (default 6667).
warning_num	Creates a warning event when this number of users are seen.

Option	Description
critical_num	Creates a critical event when this number of users are seen.

Daemons

Type	Name
Performance Collector	zencommand

(JabberMonitor) Jabber Monitor 29

The ZenPacks.zenoss.JabberMonitor ZenPack monitors the response times of Jabber instant messaging servers.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x
Required ZenPacks	ZenPacks.zenoss.JabberMonitor

Enable Monitoring

To enable monitoring:

1. Select Infrastructure from the navigation bar.
2. Click the device in the device list.

   The device overview page appears.
3. Expand Monitoring Templates in the left panel, and then select Device.
4. Select Bind Templates from the Action menu.

   The Bind Templates dialog appears.
5. Move the Jabber template from the Available list to the Selected list, and then click **Save**.

   The Jabber template is added. The system can begin collecting Jabber server metrics from the device.
6. Select the newly added template and change options as needed.

Table 30: Jabber Data Source Options

Option	Description
Timeout (seconds)	Seconds before connection times out (default: 60)
Port	The port on which the Jabber server is listening. Typically this is port 5223.

Option	Description
Send String	string to send to the server : default `<stream:stream to='${dev/id}' xmlns:stream='http://etherx.jabber.org/streams'>`
Expect String	String to expect in server response. `<stream>`

Daemons

Type	Name
Performance Collector	zencommand

(JBossMonitor) JBoss Application Server 30

The ZenPacks.zenoss.JBossMonitor ZenPack monitors JBoss application servers.

This ZenPack uses the JMX Remote API and accesses MBeans deployed within JBoss that contain performance information about the components that are being managed. The collected performance information includes: pool sizes for data sources (JDBC), Enterprise Java Beans (EJBs), message queues (JMS), threads, servlets, JSPs, and classloaders. Cache information is also accessible, providing system administrators insight into the number of hits (or misses) their cache policy has produced.

This ZenPack also aggregates individual performance metrics into higher level concepts that provide a picture of the performance of the application. Cache hits and misses are combined on the same graph to provide an overall picture of cache performance. Likewise, queue metrics are combined to show the number of messages currently on the queue, being processed, and being placed on the queue. Queue subscribers and publishers are also graphed.

Each of the individual performance metrics can be trended and predicted, and thresholds can be explicitly defined. Both the predicted thresholds and explicit thresholds inform system administrators of potential future problems before they occur. Since so much of J2EE involves "managed resources", the ability to monitor pool sizes and alert administrators prior to resources being exhausted is extremely valuable and can reduce the likelihood of a fatal outage caused by resource depletion.

Most of the metrics represent combinations of individual component metrics. For example, the Thread Pool metric represents all threads in all pools. It is possible to configure this ZenPack to perform at higher granularity and have it monitor a Thread Pool with a particular name. However, since these names are application specific we have chosen to configure this ZenPack to collect at a rather coarse-grained level by default. The installer is highly encouraged to customize and configure!

One particular monitoring template that requires end-user configuration involves Servlets. If a site to be monitored is revenue generating, and credit card submissions from the website are handled via a back-end servlet, it may be critically important to monitor the resources made available by the JBoss container to the servlet container. If the number of free spaces in the servlet pool dwindles to zero it could prevent your application from making a sale.

The following are the collected metrics for JBoss servers:

- Active Threads
- JMS Message cache memory usage
- JMS Message hits/misses
- JMS Topic/Destination queue size
- Java heap memory usage
- JCA commit, rollback, and transaction count
- JCA Connection pool in-use connections and available connections

- JCA connections created/destroyed
- JCA total connections
- JGroups cluster messages sent/received
- JGroups cluster bytes sent/received
- MBean creation/removal count
- MBean messages processed count

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.ZenJMX, ZenPacks.zenoss.JBossMonitor

Configuring JBoss to Allow JMX Queries

JBoss uses the JAVA_OPTS approach for enabling remote access to MBeans. However, it requires some additional properties. To set up your JAVA_OPTS for use in JBoss see the following code segment:

```
JAVA_OPTS="-Dcom.sun.management.jmxremote.port=12345"
JAVA_OPTS="${JAVA_OPTS} -
Dcom.sun.management.jmxremote.authenticate=false"
JAVA_OPTS="${JAVA_OPTS} -Dcom.sun.management.jmxremote.ssl=false"
JAVA_OPTS="${JAVA_OPTS} -Djboss.platform.mbeanserver"
JAVA_OPTS="${JAVA_OPTS} -
Djavax.management.builder.initial=org.jboss.system\
.server.jmx.MBeanServerBuilderImpl"
export JAVA_OPTS
```

When you start JBoss via the run.sh you must also pass the "-b 0.0.0.0" argument:

```
cd ${JBOSS_HOME}/bin
./run.sh -b 0.0.0.0
```

JMX actually uses two separate ports for MBean access: one is used for initial connection handling and authentication, and the other is used for RMI access. During the handshake between a JMX Client and the JMX Agent the agent tells the client the IP address and port number for the RMI registry. By default JBoss sets the IP address to 127.0.0.1. This works when the JMX client and the JMX agent reside on the same device, but it won't work in a distributed environment.

By passing the "-b 0.0.0.0" argument you instruct JBoss to bind to all available network ports, and this results in the JMX Agent's handshaking logic using a network reachable address when informing clients of the RMI registry hostname and port.

The `jmx-console` Web page in JBoss allows you to view the different MBeans that are available; however, this does not mean that these MBeans are available remotely. If `JConsole` can view MBeans, then so can the `zenjmx` daemon that gathers this information.

Configuring Resource Manager

All JBoss services must have a device entry under the `/Devices/Server/JBoss` device class.

Note The `zenjmx` daemon must be configured and running. For more information, see *(ZenJMX) Java Management Extensions* on page 212.

1 Navigate to the device or device class in the Resource Manager interface.

- If applying changes to a device class:
  1 Select the class in the devices hierarchy.
  2 Click **Details**.
  3 Select Configuration Properties.
- If applying changes to a device:
  1 Click the device in the device list.
  2 Select Configuration Properties.

2 Edit the appropriate configuration properties for the device or devices.

Table 31: JBoss Configuration Properties

Name	Description
zJBossJmxManagementAuthenticate	This configuration property is deprecated.
zJBossJmxManagementPassword	JMX password
zJBossJmxManagementPort	The port number used to gather JMX information
zJBossJmxManagementUsername	JMX username for authentication

3 Click **Save** to save your changes.

You will now be able to start collecting the JBoss server metrics from this device.

4 Navigate to Graphs and you should see some placeholders for graphs. After approximately fifteen minutes you should see the graphs start to become populated with information.

Note The out-of-the-box JBoss data source configuration has been defined at the macro level, but can be configured to operate on a more granular basis. For example, the Servlet Reload Count applies to all servlets in all Web applications but it could be narrowed to be Servlet /submitOrder in Web application "production server."

Change the Amount of Data Collected and Graphed

1 Navigate to the device or device class under the `/Devices/Server/JBoss` device class in the interface.
2 In the left panel, select Monitoring Templates
3 Select Bind Templates from the Action menu.
4 To add other templates and retain existing monitoring templates, hold down the control key while clicking on the original entries.

Table 32: JBoss Templates

Name	Description
JBoss Core	Core information about any JBoss server, including memory usage, threads, and uptime.

Name	Description
JBoss JCA Connection Pool	
JBoss JGroups Channel	
JBoss JMS Cache	
JBoss JMS Destination	
JBoss JMS Topic	
JBoss Message Driven EJB	

5 Click the **OK** button to save your changes.

Viewing Raw Data

For more information about how to investigate raw data returned from the application, see *Using JConsole to Query a JMX Agent* on page 220.

Daemons

Type	Name
Performance Collector	zenjmx

(JuniperMonitor) Juniper Monitor 31

The ZenPacks.zenoss.JuniperMonitor ZenPack monitors devices from Juniper Networks.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.JuniperMonitor

Configuring Juniper Devices to Allow SNMP Queries

Configure the Juniper device to allow SNMP queries from the Resource Manager server, and send SNMP v1 or SNMP v2 traps to the Resource Manager server.

Configuring Resource Manager

All Juniper devices must exist under the `/Devices/Network/Juniper` device class.

1 Navigate to the device or device class in the Resource Manager interface.

- If applying changes to a device class:
  1 Select the class in the devices hierarchy.
  2 Click **Details**.
  3 Select Configuration Properties.
- If applying changes to a device:
  1 Click the device in the device list.
  2 Select Configuration Properties.

2 Edit the appropriate configuration properties for the device or devices.

Table 33: Juniper Configuration Properties

Name	Description
zSnmpCommunity	Consult with your network administrators to determine the SNMP community permitted.
zSnmpMonitorIgnore	Set to a value of `False`.
zSnmpPort	Set the SNMP port. The default port is 161.
zSnmpVer	Set the SNMP version. Set to a value of `v2c`.

3 Click **Save** to save your changes. You will now be able to start collecting the Juniper device metrics from this device.

4 Navigate to Graphs and you should see some place holders for graphs. After approximately fifteen minutes you should see the graphs start to become populated with information.

Daemons

Type	Name
Modeler	zenmodeler
Performance Collector	zenperfsnmp

(LDAPAuthenticator) LDAP Authentication 32

The ZenPacks.zenoss.LDAPAuthenticator ZenPack enables pass-through authentication to external LDAP-based servers such as Microsoft Active Directory or OpenLDAP.

This capability allows users to sign on to the Resource Manager user interface with the same credentials they use to log in to their workstations. This saves you from having to manually create user accounts and maintain passwords in Resource Manager.

Among the benefits of using a service like LDAP to maintain user accounts and privileges are:

- Users do not have to remember another password. This decreases support and maintenance requirements.
- Centralized management of each user's privileges. This enables easier security auditing and SOX reporting.

Authentication logging is stored in the `$ZENHOME/log/event.log` file.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x
Required ZenPacks	ZenPacks.zenoss.LDAPAuthenticator

LDAP Configuration

Before configuring LDAP authentication, you should gather the following information from your LDAP or Active Directory administrator:

- Host name or IP address of an Active Directory global catalog server (for Active Directory authentication)
- Host name or IP address of an LDAP server (for other LDAP server authentication)
- User's base distinguished name (DN)
- Manager DN
- Manager password
- Groups base DN
- Optionally, list of Active Directory groups to map to Resource Manager roles

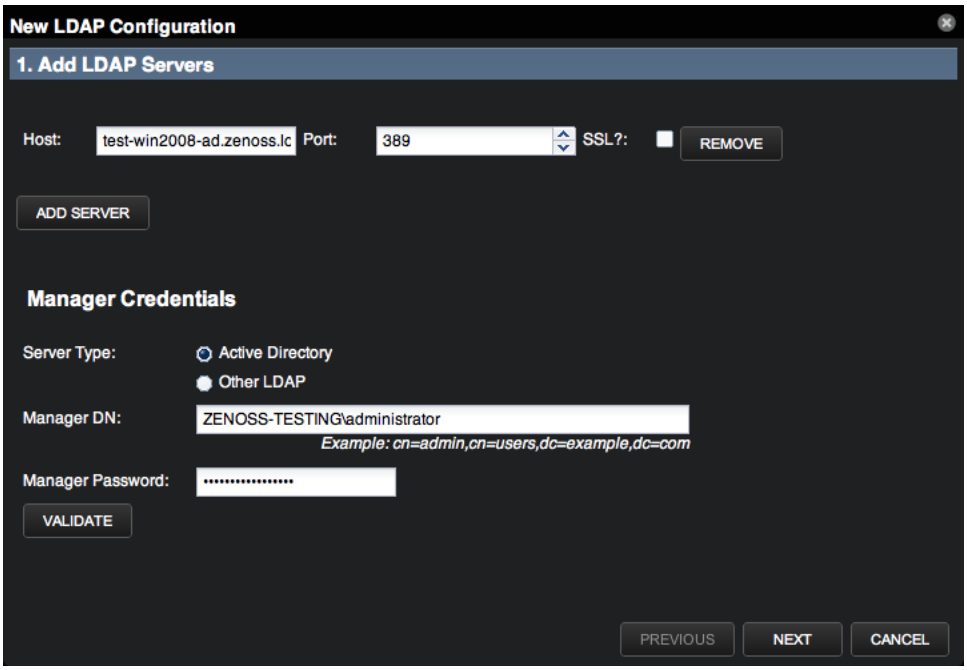
Configuring LDAP Authentication

You can configure LDAP authentication at initial setup, or from the Settings area of the interface:

- While in the setup wizard, at Step 2: Specify or Discover Devices to Monitor, click **LDAP Setup** (located at the bottom right of the wizard panel).
- From the interface, select Advanced > Settings, and then select LDAP in the left panel.

The first panel (Add LDAP Servers) of the LDAP Configuration wizard appears.

Figure 18: LDAP Configuration Wizard (Add LDAP Servers)



1 Enter information and make selections in the LDAP Servers area:

   - **Host** - Enter the host name or IP address of an Active Directory global catalog server (for Active Directory authentication) or the host name or IP address of an LDAP server (for Other LDAP server types).
   - **Port** - Optionally, change the server port number. By default, the port number is 389.
   - **SSL** - Select if using SSL. When you select this option, the default port number adjusts to 636.

2 Optionally, click **Add Server** to add another LDAP server. To remove a server from the list, click **Remove**.

3 Enter information and make selections:

   - **Server Type** - Select a server type (Active Directory or Other LDAP).
   - **Manager DN** - Enter the distinguished name of a user in the domain administrators group. An example that follows the user's base DN is:

     ```
     cn=admin,cn=users,dc=example,dc=com
     ```

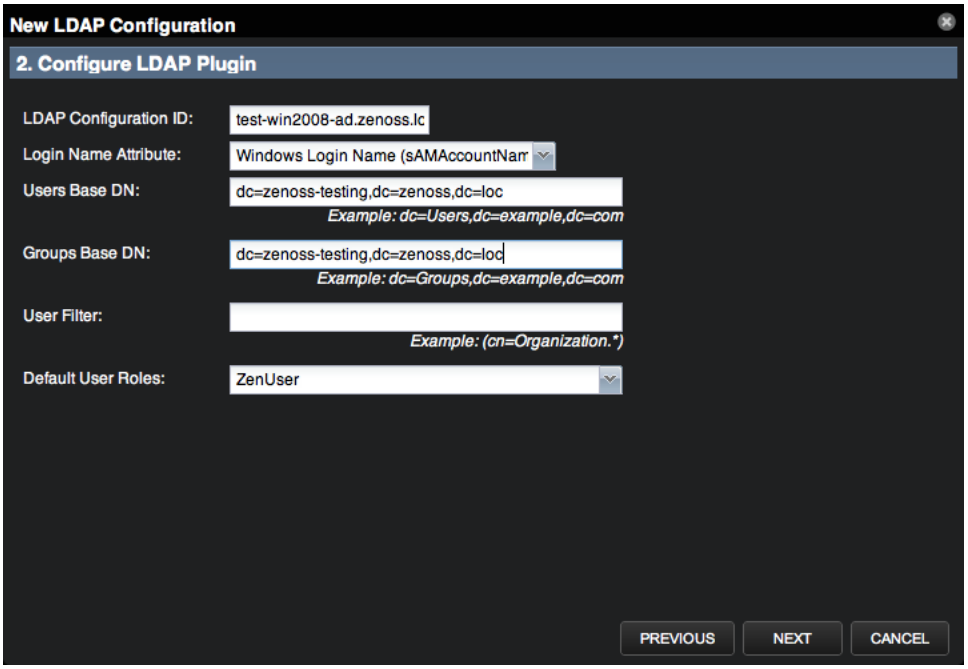
   - **Manager Password** - Enter the password for the Manager DN.

4 Click **Validate** to ensure your setup is valid.

5 Click **Next**.

The second panel (Configure LDAP Plugin) of the LDAP Configuration wizard appears.

Figure 19: LDAP Configuration Wizard (Configure LDAP Plugin)



6 Enter information and make selections:

- **Login Name Attribute** - Select the LDAP record attribute used as the user name.

  Note You can edit the list of selections by adding attributes on the Mappings page of the LDAP configuration area (Advanced > Settings > LDAP).

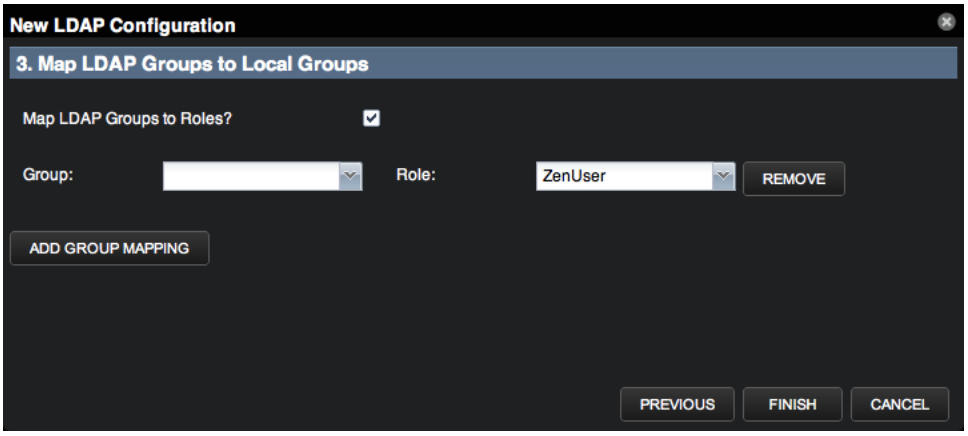
- **Users Base DN** - Enter the user's base distinguished name. For example, if your domain is ad.zenoss.com, then your user's base DN might be:

  ```
  dc=Users,dc=ad,dc=com
  ```

- **Groups Base DN** - Enter the DN for the branch of your LDAP database that contains group records. These group records are of the LDAP class "groupOfUniqueNames," and the entry CN attribute constitutes the group name.
- **User Filter** - Specify a free-form LDAP filter expression to be added to the default user search filter. The default user search filter and this additional search filter are combined as an AND expression. Records must satisfy both filters to be found using the various user searches. Any value specified in this field must follow correct LDAP search filter syntax.
- **Default User Roles** - Specify one or more roles (in a comma-delimited list) to be given to all users authenticated from your LDAP tree. Zope expects all users - anonymous as well as authenticated - to have the role Anonymous.

7 Click **Next**. The third panel (Map LDAP Groups to Local Groups) of the LDAP Configuration wizard appears.

Figure 20: LDAP Configuration Wizard (Map LDAP Groups to Local Groups)



8 Enter information and make selections:

- **Map LDAP Groups to Roles** - Select this option if you want to control user roles within the Resource Manager Web interface by using Active Directory groups, instead of controlling the roles directly from within Resource Manager.

  Note If you choose to use this option, then you should add the following groups to LDAP:

  - Resource Manager Managers
  - Resource Manager Users

- **LDAP Group** - Select the LDAP group to map to a Resource Manager role.
- **Maps to Role** - Select the Resource Manager role to map the LDAP group.

9 Optionally, click **Add Group Mapping** to map another group. To remove a mapped group, click **Remove**.

10 Click **Finish** to complete LDAP configuration.

Verifying Connectivity and Credentials Outside of Resource Manager

You can verify that your credential information is valid from the Resource Manager server by using the `ldapsearch` command. To install this command, use the following for RPM-based systems:

```
# yum -y install openldap-clients
```

as the zenoss user on the Resource Manager server:

```
ldapsearch -LLL -x -b 'BaseDN' -D 'Bind DN' -W -H ldap://LDAP_server-name
 \
"sAMAccountName=*" member
```

Configuring Local Authentication as a Fallback

You can use local authentication as a fallback in the event that the LDAP server is unreachable. The local authentication plugin is called userManager.

1 Verify that the userManager plugin is available:

   a Go to the following URL to access the Zope Management Interface (ZMI):

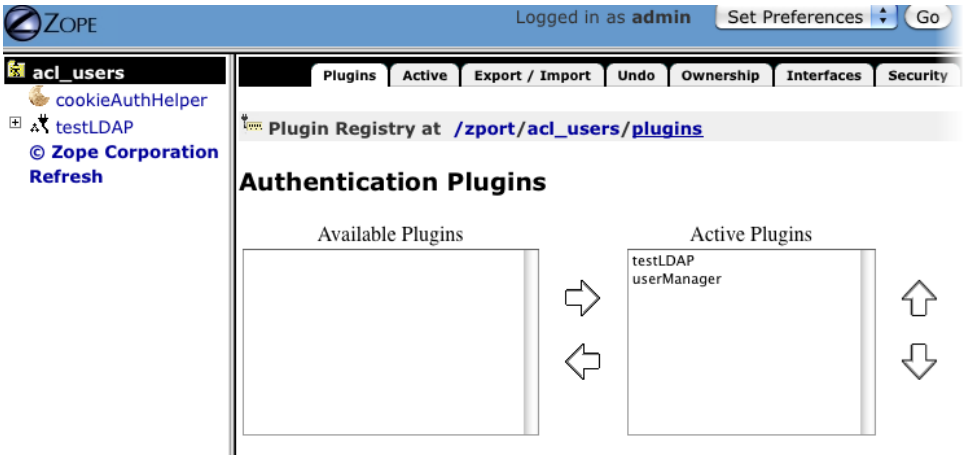
`http://YourZenossSystem:8080/zport/acl_users/manage`

b In the Name column, click **Plugins**.

c Click **Authentication Plugins**.

d Make sure that your LDAP plugin is first in the list of Active Plugins. (The userManager plugin must be below it.)

Figure 21: Authentication Plugins



2 Create a user with fallback capabilities. For example, to allow an LDAP user named "zenoss-user" to log in when the LDAP server is down:

a Go to Advanced > Settings > Users > Add New User.

b Create a user named "zenoss-user."

Note You must create this account before the user logs in with the LDAP credentials. The password defined when creating the account in Resource Manager will be valid even when the LDAP server is down.

(LDAPMonitor) LDAP Monitor 33

The ZenPacks.zenoss.LDAPMonitor ZenPack monitors the response time of Lightweight Directory Access Protocol (LDAP) servers.

The response time unit of measurement is milliseconds.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.LDAPMonitor

Enable monitoring for a device

Note The LDAPServer template must be bound to the device class or device you want to monitor.

1 Select Infrastructure from the navigation bar.
2 Click the device name in the device list.

The device overview page appears.
3 Select Configuration Properties from the left panel.
4 Modify configuration property values as needed for your environment. Check with your LDAP administrator for more information.

Table 34: LDAPServer Configuration Properties

Property	Description
zLDAPBaseDN	The Base Distinguished Name for your LDAP server. Typically this is the organization's domain name (for example, `dc=foobar,dc=com`)
zLDAPBindDN	The Distinguished Name to use for binding to the LDAP server, if authentication is required

Property	Description
zLDAPBindPassword	The password to use for binding to the LDAP server, if authentication is required

5 Click **Save**.
6 Expand Monitoring Templates, and then select Device from the left panel.
7 Select Bind Templates from the Action menu.

The Bind Templates dialog appears.
8 Add the LDAPServer template to the list of selected templates, and then click **Submit**.

The LDAPServer template is added to the list of monitoring templates.
9 Select the LDAPServer template and change options as needed.

Table 35: LDAPServer Basic Data Source Options

Option	Description
Port	The port to connect to LDAP server (default 389)
Base Distinguished Name	Defaults to `${here/zLDAPBaseDN}`
Bind Password	Defaults to `${here/zLDAPBindPassword}`
Use SSL	Use SSL for the connection

Note If your LDAP servers require SSL or a custom port, select the ldap data source, and then change the Use SSL and Port fields as needed.

10 Validate your configuration by running `zencommand`. Verify that the `check_ldap` or `check_ldaps` command correctly connects to your LDAP server:

```
zencommand run -v10 -d yourdevicenamehere
```

Daemons

Type	Name
Performance Collector	`zencommand`

(LinuxMonitor) Linux Monitor 34

The ZenPacks.zenoss.LinuxMonitor ZenPack demonstrates how to develop new plugins that collect performance data using Secure Shell.

This ZenPack demonstrates the Secure Shell (SSH) features, and enables modeling and monitoring several types of device components for devices placed in the `/Server/SSH/Linux` device class by running commands and parsing the output. Parsing of command output is performed on the Resource Manager server or on a distributed collector. The account used to monitor the device does not require root access or special privileges.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.4 or higher
Required ZenPacks	ZenPacks.zenoss.LinuxMonitor

Set Linux Server Monitoring Credentials

All Linux servers must have a device entry in an organizer below the `/Devices/Server/SSH/Linux` device class.

Note The SSH monitoring feature will attempt to use key-based authentication before using a configuration properties password value.

1. Select Infrastructure from the navigation bar.
2. Click the device name in the device list.

   The device overview page appears.
3. Select Configuration Properties from the left panel.
4. Verify the credentials for the service account.

Table 36: Linux Configuration Properties

Name	Description
zCommandUsername	Linux user with privileges to gather performance information.
zCommandPassword	Password for the Linux user.

Add a Linux Server

The following procedure assumes that credentials have been set.

1 Select Infrastructure from the navigation bar.
2 Select Add a Single Device from the Add Device list of options.

The Add a Single Device dialog appears.
3 Enter the following information in the dialog:

Table 37: Adding Linux Device Details

Name	Description
Name or IP	Linux host to model.
Device Class	`/Server/SSH/Linux`
Model Device	Select this option unless adding a device with a user name and password different than found in the device class. If you do not select this option, then you must add the credentials (see) and then manually model the device.

4 Click **Add**.

Daemons

Type	Name
Modeler	`zenmodeler`
Performance Collector	`zencommand`

(Microsoft.Windows) Microsoft Windows 35

The ZenPacks.zenoss.Microsoft.Windows ZenPack monitors Microsoft Windows systems and services through the Windows Remote Management (WinRM) and Windows Remote Shell (WinRS) interfaces.

Monitoring data is collected from the Windows Management Instrumentation (WMI) and Windows Reliability and Performance Monitor (Perfmon) services.

Note For Windows systems that support WinRM, this ZenPack replaces the following, deprecated ZenPacks.

- ZenPacks.zenoss.ActiveDirectory
- ZenPacks.zenoss.IISMonitor
- ZenPacks.zenoss.MSExchange
- ZenPacks.zenoss.MSMQMonitor
- ZenPacks.zenoss.MSSQLServer
- ZenPacks.zenoss.PySamba
- ZenPacks.zenoss.WindowsMonitor

However, installing this ZenPack does not automatically migrate Windows systems or disable monitoring through the deprecated ZenPacks. For more information about transitioning to this ZenPack, see *Transitioning to this ZenPack* on page 133.

Discovery and modeling

The components. properties, and relationships that this ZenPack discovers and updates during modeling. Discovery and modeling are enabled by specifying server addresses, usernames, and passwords.

Server (Device)

Attributes: Name, Contact, Description, Serial Number, Tag, Hardware Model, Total Memory, Total Swap, Operating System, Cluster

Cluster (Device)

Attributes: Name, Contact, Description, Total Memory, Total Swap, Operating System, Member Servers

Processors

Attributes: Name, Description, Model, Socket, Cores, Threads, Clock Speed, External Speed, Voltage, L1 Cache Size, L2 Cache Size and Speed, L3 Cache Size and Speed

File System

Attributes: Mount Point, Status, Storage Device, Type, Block Size, Total Blocks, Total Bytes, Maximum Name Length

Interfaces

Attributes: Name, Description, MAC Address, MTU, Speed, Duplex, Type, Administrative Status, Operational Status, IP Addresses

Network Routes

Attributes: Destination, Next Hop, Interface, Protocol, Type

Process Sets

Attributes: Name, Recent Matches, Process Class

Software

Attributes: Name, Vendor, Installation Date

Services

Attributes: Name, Display Name, Start Mode, Account

Cluster Services

Attributes: Name, Core Group, Owner Node, State, Description, Priority

Relationships: Cluster Resources

Cluster Resources

Attributes: Name, Owner Node, Description, Owner Group, State

Relationships: Cluster Service

IIS Sites

Attributes: Name, Status, App Pool

SQL Server Instances

Attributes: Name

Relationships: SQL Server Databases

SQL Server Databases

Attributes: Name, Version, Owner, Last Backup, Last Log Backup, Accessible, Collation, Creation Date, Default File Group, Primary File Path

Relationships: SQL Server Instance

SQL Server Backups

Attributes: Name, Device Type, Physical Allocation, Status

Relationships: SQL Server Instance

SQL Server Jobs

Attributes: Name, Job ID, Description, Enabled, Date Created, Username

Relationships: SQL Server Instance

Performance Monitoring

Perfmon counter data are collected through the PowerShell Get-Counter cmdlet, within a remote shell (WinRS). To collect data from any other Perfmon counter, simply add the counter to the appropriate Resource Manager monitoring template.

Note The following Processes metrics (Win32_PerfFormattedData_PerfProc_Process) are collected directly through Windows Management Instrumentation (WMI).

- PercentProcessorTime
- WorkingSet
- WorkingSetPrivate (not available on Windows 2003)

Device metrics

The following metrics are collected from the `Devices` Perfmon counter.

- \Memory\Available bytes
- \Memory\Committed Bytes
- \Memory\Pages Input/sec
- \Memory\Pages Output/sec
- \Paging File(_Total)\% Usage
- \Processor(_Total)\% Privileged Time
- \Processor(_Total)\% Processor Time
- \Processor(_Total)\% User Time
- \System\System Up Time

Active Directory metrics

The following metrics are collected from the `Active Directory` Perfmon counter.

- \NTDS\DS Client Binds/sec
- \NTDS\DS Directory Reads/sec
- \NTDS\DS Directory Searches/sec
- \NTDS\DS Directory Writes/sec
- \NTDS\DS Monitor List Size
- \NTDS\DS Name Cache hit rate
- \NTDS\DS Notify Queue Size
- \NTDS\DS Search sub-operations/sec
- \NTDS\DS Server Binds/sec
- \NTDS\DS Server Name Translations/sec
- \NTDS\DS Threads in Use
- \NTDS\KDC AS Requests
- \NTDS\KDC TGS Requests
- \NTDS\Kerberos Authentications
- \NTDS\LDAP Active Threads
- \NTDS\LDAP Bind Time
- \NTDS\LDAP Client Sessions
- \NTDS\LDAP Closed Connections/sec
- \NTDS\LDAP New Connections/sec
- \NTDS\LDAP New SSL Connections/sec
- \NTDS\LDAP Searches/sec
- \NTDS\LDAP Successful Binds/sec
- \NTDS\LDAP UDP operations/sec
- \NTDS\LDAP Writes/sec
- \NTDS\NTLM Authentications
- \NTDS\DS Client Binds/sec
- \NTDS\DS Directory Reads/sec
- \NTDS\DS Directory Searches/sec
- \NTDS\DS Directory Writes/sec
- \NTDS\DS Monitor List Size
- \NTDS\DS Name Cache hit rate
- \NTDS\DS Notify Queue Size

- \NTDS\DS Search sub-operations/sec
- \NTDS\DS Server Binds/sec
- \NTDS\DS Server Name Translations/sec
- \NTDS\DS Threads in Use
- \NTDS\LDAP Active Threads
- \NTDS\LDAP Bind Time
- \NTDS\LDAP Client Sessions
- \NTDS\LDAP Closed Connections/sec
- \NTDS\LDAP New Connections/sec
- \NTDS\LDAP New SSL Connections/sec
- \NTDS\LDAP Searches/sec
- \NTDS\LDAP Successful Binds/sec
- \NTDS\LDAP UDP operations/sec
- \NTDS\LDAP Writes/sec
- \DirectoryServices(NTDS)\DS Client Binds/sec
- \DirectoryServices(NTDS)\DS Directory Reads/sec
- \DirectoryServices(NTDS)\DS Directory Searches/sec
- \DirectoryServices(NTDS)\DS Directory Writes/sec
- \DirectoryServices(NTDS)\DS Monitor List Size
- \DirectoryServices(NTDS)\DS Name Cache hit rate
- \DirectoryServices(NTDS)\DS Notify Queue Size
- \DirectoryServices(NTDS)\DS Search sub-operations/sec
- \DirectoryServices(NTDS)\DS Server Binds/sec
- \DirectoryServices(NTDS)\DS Server Name Translations/sec
- \DirectoryServices(NTDS)\DS Threads in Use
- \DirectoryServices(NTDS)\LDAP Active Threads
- \DirectoryServices(NTDS)\LDAP Bind Time
- \DirectoryServices(NTDS)\LDAP Client Sessions
- \DirectoryServices(NTDS)\LDAP Closed Connections/sec
- \DirectoryServices(NTDS)\LDAP New Connections/sec
- \DirectoryServices(NTDS)\LDAP New SSL Connections/sec
- \DirectoryServices(NTDS)\LDAP Searches/sec
- \DirectoryServices(NTDS)\LDAP Successful Binds/sec
- \DirectoryServices(NTDS)\LDAP UDP operations/sec
- \DirectoryServices(NTDS)\LDAP Writes/sec

Exchange metrics

The following metrics are collected from the `Exchange` Perfmon counter.

- \MSExchangeIS Mailbox(_Total)\Folder opens/sec
- \MSExchangeIS Mailbox(_Total)\Local delivery rate
- \MSExchangeIS Mailbox(_Total)\Message Opens/sec
- \MSExchangeIS\RPC Averaged Latency
- \MSExchangeIS\RPC Operations/sec
- \MSExchangeIS\RPC Requests
- \SMTP Server(_Total)\Local Queue Length
- \SMTP Server(_Total)\Messages Delivered/sec
- \MSExchangeTransport Queues(_Total)\Active Mailbox Delivery Queue Length
- \MSExchangeTransport Queues(_Total)\Messages Completed Delivery Per Second

- \MSExchangeIS Mailbox(_Total)\Folder opens/sec
- \MSExchangeIS Mailbox(_Total)\Local delivery rate
- \MSExchangeIS Mailbox(_Total)\Message Opens/sec
- \MSExchangeIS\RPC Averaged Latency
- \MSExchangeIS\RPC Operations/sec
- \MSExchangeIS\RPC Requests

IIS metrics

The following metrics are collected from the `IIS` Perfmon counter.

- \Web Service(_Total)\Bytes Received/sec
- \Web Service(_Total)\Bytes Sent/sec
- \Web Service(_Total)\CGI Requests/sec
- \Web Service(_Total)\Connection Attempts/sec
- \Web Service(_Total)\Copy Requests/sec
- \Web Service(_Total)\Delete Requests/sec
- \Web Service(_Total)\Files Received/sec
- \Web Service(_Total)\Files Sent/sec
- \Web Service(_Total)\Get Requests/sec
- \Web Service(_Total)\Head Requests/sec
- \Web Service(_Total)\ISAPI Extension Requests/sec
- \Web Service(_Total)\Lock Requests/sec
- \Web Service(_Total)\Mkcol Requests/sec
- \Web Service(_Total)\Move Requests/sec
- \Web Service(_Total)\Options Requests/sec
- \Web Service(_Total)\Other Request Methods/sec
- \Web Service(_Total)\Post Requests/sec
- \Web Service(_Total)\Propfind Requests/sec
- \Web Service(_Total)\Proppatch Requests/sec
- \Web Service(_Total)\Put Requests/sec
- \Web Service(_Total)\Search Requests/sec
- \Web Service(_Total)\Trace Requests/sec
- \Web Service(_Total)\Unlock Requests/sec

IIS Sites metrics

The following metrics are collected from the `IIS Sites` Perfmon counter.

Note To use these metrics, install the IIS 6 Management Compatibility module on the servers to monitor.

- \Bytes Received/sec
- \Bytes Sent/sec
- \CGI Requests/sec
- \Connection Attempts/sec
- \Copy Requests/sec
- \Connection Attempts/sec
- \Delete Requests/sec
- \Files Received/sec
- \Files Sent/sec
- \Get Requests/sec

- \Head Requests/sec
- \ISAPI Extension Requests/sec
- \Lock Requests/sec
- \Mkcol Requests/sec
- \Move Requests/sec
- \Options Requests/sec
- \Other Request Methods/sec
- \Post Requests/sec
- \Propfind Requests/sec
- \Proppatch Requests/sec
- \Put Requests/sec
- \Search Requests/sec
- \Trace Requests/sec
- \Unlock Requests/sec

SQLServer metrics

The following metrics are collected from the `SQLServer` Perfmon counter.

- \SQLServer:Access Methods\Full Scans/sec
- \SQLServer:Buffer Manager\Buffer cache hit ratio
- \SQLServer:Buffer Manager\Free pages
- \SQLServer:Databases(_Total)\Data File(s) Size (KB)
- \SQLServer:General Statistics\User Connections
- \SQLServer:Latches\Latch Waits/sec
- \SQLServer:Locks(_Total)\Average Wait Time (ms)
- \SQLServer:Locks(_Total)\Lock Requests/sec
- \SQLServer:Locks(_Total)\Number of Deadlocks/sec
- \SQLServer:SQL Statistics\Batch Requests/sec

File Systems metrics

The following metrics are collected from the `File Systems` Perfmon counter.

- \Disk Read Bytes/sec
- \% Disk Read Time
- \Disk Write Bytes/sec
- \% Disk Write Time
- \Free Megabytes

Interfaces metrics

The following metrics are collected from the `Interfaces` Perfmon counter.

- \Bytes Received/sec
- \Bytes Sent/sec
- \Packets Received Errors
- \Packets Received/sec
- \Packets Outbound Errors
- \Packets Sent/sec

Event Management

Events are collected from Windows event logs through a WinRM subscription. Information encoded in Windows event classes is used to populate the following, standard Zenoss event fields.

- device
- component
- summary
- severity
- eventClassKey (for mapping specific event types)
- eventKey (for de-duplication and auto-clear fingerprinting)

Events collected in this manner are timestamped with the time from the Windows event log, not the collection time.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x
ZenPacks	ZenPacks.zenoss.PythonCollector, ZenPacks.zenoss.Microsoft.Windows
RPM Packages	krb5-workstation
Windows management	WinRM 2.0. (Windows Server 2003 with SP1 and all subsequent releases of Windows Server include WinRM 2.0.)
IIS management	To enable the IIS Sites metrics, install the IIS 6 Management Compatibility module on the Windows systems to monitor.

Installing Kerberos authentication

To use this ZenPack, Kerberos authentication must be installed on all Resource Manager hosts (master, collector, and hub hosts).

1. Log in to a Resource Manager host as `root`, or as a user with superuser privileges.
2. Determine whether the Kerberos authentication package is installed.
   ```
   rpm -qa | grep -i krb5-workstation
   ```
   - If the command returns a result, Kerberos authentication is installed. Discontinue this procedure.
   - If the command does not return a result, proceed to the next step.
3. Install Kerberos authentication.
   ```
   yum -y install krb5-workstation
   ```

Adding a Windows device

Follow these steps to add a Windows device through the Resource Manager user interface.

1. Log in to Resource Manager user interface as a user with ZenManager or Manager privileges.
2. Navigate to the **INFRASTRUCTURE** page.
3. In the left column, select the **/Server/Microsoft/Windows** class, and then click **DETAILS**.
4. Select **Configuration Properties**, and then provide values for the `zWinRMUser` and `zWinRMPassword` properties.
5. In the left column, click **SEE ALL**.
6. From the the **Add** menu, select **Add a Single Device**.
7. Complete the form with the information for the Windows device to add.

The value of the **Name or IP** field must be resolvable and accessible from the collector host specified in the **Collector** field.

8 Click **ADD**.

Adding multiple Windows devices

Follow these steps to add multiple Windows devices with the `zenbatchload` command.

1 Log in to the Resource Manager master host as `zenoss`.

2 Create a text file with content similar to the following example.

```
/Devices/Server/Microsoft/Windows
FQDN-or-IP-address zWinRMUser="Administrator",
 zWinRMPassword="Password"
FQDN-or-IP-address zWinRMUser="Administrator",
 zWinRMPassword="Password"
FQDN-or-IP-address zWinRMUser="Administrator",
 zWinRMPassword="Password"
```

Replace *FQDN-or-IP-address* with the fully-qualified domain name or IP address of the Windows host to add, and replace *Password* with the Administrator's password for each host.

3 Load the devices into Resource Manager. Replace *Filename* with the name of the file created in the preceding step.

```
zenbatchload Filename
```

Configuring systems for monitoring

You may configure the Windows systems to monitor with this ZenPack collectively or individually.

- To configure systems collectively, use Windows Group Policy.

  Note This option may require configuration steps on each Windows system to monitor.

- To configure systems individually, use the procedure in this section.

Note To enable the IIS Sites metrics, install the IIS 6 Management Compatibility module on the Windows systems to monitor.

Authentication and transport options

For authentication and transport, you may configure one of the options in the following list. This ZenPack supports all four options.

Basic authentication, HTTP transport

This option is the least secure and is not recommended. Usernames, passwords, and payloads are transmitted in clear text.

Basic authentication, HTTPS transport

This option encrypts usernames, passwords, and payloads at the transport layer, with SSL. The Windows systems to monitor must be configured to support HTTPS individually.

Kerberos authentication, HTTP transport

This option encrypts usernames and passwords (at the application layer) but payloads are not encrypted.

Kerberos authentication, HTTPS transport

This option is the most secure and is recommended. Usernames, passwords, and payloads are all encrypted, first at the application layer, and again at the transport layer. The Windows systems to monitor must be configured to support HTTPS individually.

Note NTLMv2 is not supported, and cannot be configured to work with this ZenPack.

Configuring systems with Windows Group Policy

Windows Remote Management (WinRM)

The `Computer Configuration\Policies\Administrative Templates\Windows Components\Windows Remote Management` category includes the WinRM Client and WinRM Service sub-categories. For WinRM Client, no policy changes are required. The default policies provide the needed support.

The following table shows the WinRM Service policies to set for the desired authentication and transport option.

Table 38: WinRM Service policy options

Policy	Authentication/Transport option			
	Basic/HTTP	**Basic/HTTPS**	**Kerberos/ HTTP**	**Kerberos/ HTTPS**
Allow remote server management through WinRm	Enabled	Enabled	Enabled	Enabled
Allow unencrypted Traffic	Enabled	Disabled	Enabled	Disabled
Allow Basic authentication	Enabled	Enabled	Disabled	Disabled

For more information about configuring HTTPS, see *Configuring HTTPS* on page 132.

Windows Remote Shell (WinRS)

Configure the following policies in the `Computer Configuration\Policies\Administrative Templates\Windows Components\Windows Remote Shell` category.

- Policy: *Allow Remote Shell Access*

  Choose the default setting, which allows remote shell connections.
- Policy: *Specify maximum number of processes per Shell*

  Set the value to 4294967295.

  Note This is the maximum value. The default (5) is inadequate, because Resource Manager opens concurrent requests for each WQL query and set of Perfmon counters. If the maximum value is inappropriate for your environment, 50 should be adequate.
- Policy: *Specify maximum number of remote shells per user*

  Set the value to 2147483647.

  Note This is the maximum value. The default (5) is inadequate, because Resource Manager opens concurrent requests for each WQL query and set of Perfmon counters. If the maximum value is inappropriate for your environment, 50 should be adequate.

- Policy: *Specify Shell Timeout*

  Set the value to 7200000.

Configuring HTTPS

Currently, HTTPS must be configured on each Windows system to monitor. Zenoss is testing several options for automating the task, but none are available for production use.

To successfully encrypt the payload between Resource Manager and Windows clients, you must *install a Server Authentication certificate on the client system*. Once the correct certificate is installed, change the client's `zWinScheme` property to HTTPS and the `zWinRMPort` property to 5986.

If the preceding steps are unsuccessful, verify that the appropriate Service Principal Name (SPN) record exists for Kerberos authentication. Log in to any Active Directory system and enter the following command. Replace *Hostname* with the hostname portion of the client system's fully-qualified domain name (FQDN).

```
setspn -l Hostname
```

If the preceding command returns a hostname record that does not start with `HTTPS/`, use the following command to create the record. Replace *FQDN* with the fully-qualified domain name of the client system.

```
setspn -s HTTPS/FQDN Hostname
```

Configuring systems individually

You may configure the Windows systems to monitor individually, without Windows Group Policy.

Note If you choose the default WinRM configuration, supply Kerberos authentication settings in `zProperties`. The Kerberos authentication process requires a Key Distribution Center (KDC), which acts as both an authentication server and a ticket granting server. Microsoft Active Directory is a KDC, so the `zWinKDC` value must be set to the IP address of the Active Directory server, and Resource Manager collectors must be able to send TCP/IP packets to it. The value of `zWinRMUserName` must include a fully qualified domain name (for example, `jsmith@zenoss.com`) and `zWinRMPassword` must include the password of the user account.

Follow these steps to enable this ZenPack to monitor a Windows system.

1. Log on to the Windows system to enable, and open the **Windows Firewall with Advanced Security** utility, or its equivalent.
2. Open port 5985 (for HTTP) or 5986 (for HTTPS) for WinRM, and then close the utility.
3. Start a command window as Administrator.
4. Configure the system to accept WS-Management requests from other systems.
   ```
   winrm quickconfig
   ```
5. Configure the maximum number of concurrent operations per user.
   ```
   winrm s winrm/config/service
   @{MaxConcurrentOperationsPerUser="4294967295"}
   ```

   Note This is the maximum value. The default (5) is inadequate, because Resource Manager opens concurrent requests for each WQL query and set of Perfmon counters. If the maximum value is inappropriate for your environment, 50 should be adequate.

   ---

6. Configure the maximum number of shells per user.
   ```
   winrm s winrm/config/winrs @{MaxShellsPerUser="2147483647"}
   ```

Note This is the maximum value. The default (5) is inadequate, because Resource Manager opens concurrent requests for each WQL query and set of Perfmon counters. If the maximum value is inappropriate for your environment, 50 should be adequate.

7 Configure the idle timeout.
```
winrm s winrm/config/winrs @{IdleTimeout="7200000"}
```
8 Optional: Configure HTTP. The default is HTTPS.
```
winrm s winrm/config/service @{AllowUnencrypted="true"}
```
For more information about configuring HTTPS, see *Configuring HTTPS* on page 132.

9 Optional: Configure Basic Authentication. The default is Kerberos.
```
winrm s winrm/config/service/auth @{Basic="true"}
```

Transitioning to this ZenPack

This ZenPack provides a superset of the functionality provided in earlier Windows monitoring ZenPacks, with better performance and reliability.

Note The new and old Windows monitoring ZenPacks co-exist gracefully—no automatic migration is attempted, and all history is preserved.

Zenoss recommends transitioning to this ZenPack as soon as possible in all but the following situations:

- You must continue to monitor Windows systems that do not support WinRM 2.0.

  At some point, Zenoss will discontinue support for the deprecated ZenPacks, so moving older Windows systems to a version that supports WinRM 2.0 is encouraged.
- You must maintain an unbroken history of monitoring, using exactly the same metrics, gathered in exactly the same way.

  Testing or regulatory requirements may prevent an immediate migration. However, no monitoring history is discarded when this ZenPack is enabled. You may maintain the RRD files created with the deprecated ZenPacks alongside the new ones, and use them to examine the complete monitoring history.

To determine whether the deprecated Windows monitoring ZenPacks are installed, log in to the broswer interface, and navigate to ADVANCED > ZenPacks. Look for the following ZenPacks.

- ZenPacks.zenoss.ActiveDirectory
- ZenPacks.zenoss.IISMonitor
- ZenPacks.zenoss.MSExchange
- ZenPacks.zenoss.MSMQMonitor
- ZenPacks.zenoss.MSSQLServer
- ZenPacks.zenoss.PySamba
- ZenPacks.zenoss.WindowsMonitor

Moving Windows systems to this ZenPack

1 Log in to Resource Manager user interface as a user with ZenManager or Manager privileges.
2 Navigate to the **INFRASTRUCTURE** page.
3 In the left column, select the `/Server/Windows/WMI` device class.
4 In the device list, select a Windows system to remove from the class.
5 From the bottom of the left column, click the Action button, and then select **Delete Device...**.
6 In the **Delete Device** dialog, click **SUBMIT**.

Add the deleted Windows device to the `/Server/Microsoft/Windows` class. For more information, see *Adding a Windows device* on page 129.

Using this ZenPack with the deprecated ZenPacks

You may use this ZenPack with the deprecated Windows monitoring ZenPacks

- by adding some Windows systems to the `/Server/Microsoft/Windows` device class and leaving other systems in the `/Server/Windows/WMI` device class
- by adding a subset of this ZenPack, such as the Windows Shell data source, to devices in the `/Server/Windows/WMI` device class

To add a subset of this ZenPack to a device, follow these steps.

1. In the `/Server/Windows/WMI` device class, select the device to modify.
2. Verify that the following properties are configured.

   zWinUser

   Use the DOMAIN\Username format required for DCOM/RPC collection.

   zWinPassword

   The password for the zWinUser account.

   zWinRMUser

   Use the username@example.com format required for WinRM collection.

   zWinRMPassword

   The password for the zWinRMUser account.
3. Create a monitoring template containing a Windows Shell datasource, and then bind it to the device.

Installed items

This ZenPack installs the following items.

Device classes

/Server/Microsoft

/Server/Microsoft/Cluster

/Server/Microsoft/Windows

Configuration properties

zWinRMUser

zWinRMPassword

zWinRMPort

zDBInstances

zDBInstancesPassword

zWinKDC

zWinKeyTabFilePath

zWinScheme

zWinPerfmonInterval

Modeler plugins

zenoss.winrm.CPUs

zenoss.winrm.FileSystems

zenoss.winrm.IIS

zenoss.winrm.Interfaces

zenoss.winrm.OperatingSystem
zenoss.winrm.Processes
zenoss.winrm.Routes
zenoss.winrm.Services
zenoss.winrm.Software
zenoss.winrm.WinCluster
zenoss.winrm.WinMSSQL

Datasource types

Windows EventLog
Windows IIS Site
Windows Perfmon
Windows Process
Windows Service
Windows Shell

Monitoring templates

Device (in /Server/Microsoft)
FileSystem (in /Server/Microsoft)
ethernetCsmacd (in /Server/Microsoft)
OSProcess (in /Server/Microsoft)
OSProcess-2003 (in /Server/Microsoft)
WinService (in /Server/Microsoft)
Active Directory (in /Server/Microsoft)
Active Directory 2008 (in /Server/Microsoft)
Active Directory 2008R2 (in /Server/Microsoft)
IIS (in /Server/Microsoft)
IISADMIN (in /Server/Microsoft)
IISSites (in /Server/Microsoft)
MSExchangeIS (in /Server/Microsoft)
MSExchangeIS 2007 (in /Server/Microsoft)
MSSQLServer (in /Server/Microsoft)
WinDatabase (in /Server/Microsoft)
Cluster (in /Server/Microsoft)
ClusterService (in /Server/Microsoft/Cluster)
ClusterResource (in /Server/Microsoft/Cluster)

(MSExchange) Microsoft Exchange 36

The ZenPacks.zenoss.MSExchange ZenPack uses WMI to monitor Microsoft Exchange and related services.

Note This ZenPack is deprecated; see *(Microsoft.Windows) Microsoft Windows* on page 123.

The ZenPack enables users to view graphs based on MS Exchange Performance Counters and to monitor processes related to MS Exchange.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x
Required ZenPacks	ZenPacks.zenoss.WindowsMonitor, ZenPacks.zenoss.MSExchange

Enable Monitoring

All MS Exchange services must have a device entry under the `/Devices/Server/Windows/MSExchange` device class. In addition, verify that your Resource Manager Windows service account has access to the MS Exchange service.

1. Navigate to the device or device class in the Resource Manager interface.
   - If applying changes to a device class:
     1. Select the class in the devices hierarchy.
     2. Click **Details**.
     3. Select Configuration Properties.
   - If applying changes to a device:
     1. Click the device in the device list.
     2. Select Configuration Properties.
2. Verify the credentials for the service account to access the service.

Table 39: MS Exchange Configuration Properties

Name	Description
zWinUser	Windows user with privileges to gather performance information.
zWinPassword	Password for the above user.

3 Click **Save** to save your changes.

You will now be able to start collecting the MS Exchange server metrics from this device.

4 Navigate to Graphs and you should see some placeholders for graphs. After approximately fifteen minutes you should see the graphs start to become populated with information.

Note For more information about user credentials and troubleshooting WMI connections, see *WindowsMonitor (Microsoft Windows)* on page 197.

Daemons

Type	Name
Performance Collector	`zenwinperf`

(MSMQMonitor) Microsoft Message Queueing 37

The ZenPacks.zenoss.MSMQMonitor ZenPack uses WMI to automatically discover Microsoft Message Queuing (MSMQ) queues, and monitor the number of messages queued in each.

Note This ZenPack is deprecated; see *(Microsoft.Windows) Microsoft Windows* on page 123.

The following description of Microsoft Message Queuing (MSMQ) can be found on Microsoft's MSMQ product page.

"Microsoft Message Queuing (MSMQ) technology enables applications running at different times to communicate across heterogeneous networks and systems that may be temporarily offline. MSMQ provides guaranteed message delivery, efficient routing, security, and priority-based messaging. It can be used to implement solutions for both asynchronous and synchronous messaging scenarios."

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x
Required ZenPacks	ZenPacks.zenoss.WindowsMonitor, ZenPacks.zenoss.MSMQMonitor

Configuration

To monitor MSMQ queues, set up proper credentials so that Resource Manager can remotely monitor the target Windows servers. For more information, refer to *WindowsMonitor (Microsoft Windows)* on page 197.

This ZenPack supports two approaches to enable MSMQ queue monitoring, as detailed in the next sections.

Automatically Monitor Queues on All Servers

The easiest way to configure Resource Manager to monitor your queues is to enable queue discovery for the entire **/Server/Windows** device class. Within 12 hours Resource Manager will have automatically discovered all of the queues available to be monitored and begun monitoring how many messages are in each queue and creating threshold events if they exceed 10,000 messages.

Perform the following steps to enable queue discovery for all Windows servers.

1. Navigate to the **/Server/Windows** device class.
2. Click **Details**.
3. Select Modeler Plugins from the left panel.
4. Click **Add Fields**.
5. Drag **zenoss.wmi.MSMQQueueMap** from the available fields to the list of plugins.
6. Click **Save**.
7. Wait about 12 hours for all Windows servers to be remodeled.

Monitor Queues on Specific Servers

If you do not want Resource Manager automatically monitoring queues on all of your Windows servers and would rather point it to specific servers you can do so by performing the following steps on each server you're interested in.

1. Navigate to the device.
2. Select Modeler Plugins from the left panel.
3. Click **Add Fields**.
4. Drag **zenoss.wmi.MSMQQueueMap** from the available fields to the list of plugins.
5. Click **Save**.
6. Select Model Device from the Action menu.

Fine-Tuning Queue Monitoring

By default Resource Manager will automatically monitor all queues on a server that is running the MSMQ services. Each queue will also have a default 10,000 maximum threshold applied to it. This means that an event will be created when the number of messages in a single queue exceeds 10,000.

Note By default queues with names beginning with `tcp` will not be discovered. You can change this behavior with the zMSMQIgnoreQueues property. This property is a regular expression and any queues that match it will not be discovered.

You can change the maximum messages threshold on a per-queue basis by changing the **Queues Messages Threshold** property. Leaving this value blank will have the result of no threshold being applied.

Daemons

Type	Name
Modeler	`zenmodeler`
Performance Collector	`zenwinperf`

(MSSQLServer) Microsoft SQL Server 38

The ZenPacks.zenoss.MSSQLServer ZenPack uses WMI to monitor Microsoft SQL Server and its related services.

Note This ZenPack is deprecated; see *(Microsoft.Windows) Microsoft Windows* on page 123.

The ZenPack enables users to view graphs based on Microsoft SQL Server Performance Counters and to monitor processes related to SQL Server.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x
Required ZenPacks	ZenPacks.zenoss.WindowsMonitor, ZenPacks.zenoss.MSSQLServer

Enable Monitoring

All MS SQL Server services must have a device entry under the `/Devices/Server/Windows/MSSQLServer` device class. In addition, verify that your Resource Manager Windows service account has access to the MS SQL Server service.

1. Navigate to the device or device class in the Resource Manager interface.
   - If applying changes to a device class:
     1. Select the class in the devices hierarchy.
     2. Click **Details**.
     3. Select Configuration Properties.
   - If applying changes to a device:
     1. Click the device in the device list.
     2. Select Configuration Properties.
2. Verify the credentials for the service account to access the service.

Table 40: MS SQL Server Configuration Properties

Name	Description
zWinUser	Windows user with privileges to gather performance information.
zWinPassword	Password for the above user.

3 Click **Save** to save your changes.

You will now be able to start collecting the MS SQL Server server metrics from this device.

4 Navigate to Graphs to see placeholders for graphs. After approximately fifteen minutes, the graphs start to become populated with information.

Note For more information about user credentials and troubleshooting WMI connections, see *WindowsMonitor (Microsoft Windows)* on page 197.

Collecting Information from Non-Default Microsoft SQL Server Instances

The default Microsoft SQL Sever instance is SQLServer. The monitoring template delivered with this ZenPack uses this default instance to gather performance metrics. If you use a non-default SQL Server instance, then Resource Manager does not automatically find and gather information about it.

To enable Resource Manager to monitor a non-default instance, you must override the monitoring template:

1 From Infrastructure > Devices, click the device on which you want to override the template.
2 Under Monitoring Templates, select the MSSQLServer template.
3 From the Action menu, select Override Template Here.

The Override Templates dialog appears.

4 Select the MSSQLServer template in the list, and then click **Submit**.

The template redisplays in the left panel, now identified as "Locally Defined."

5 For each of the data sources in the Data Sources area, perform these steps:

   a Double-click the data source to edit it.
   b In the Perf Counter field, change the text "\SQLServer:" to "*MyInstance*:" (where *MyInstance* is the name of the Microsoft SQL Server database instance name.
   c Click **Save**.

6 Remodel the device.

Daemons

Type	Name
Performance Collector	`zenwinperf`

(MultiRealmIP) Multi-Realm IP Networks 39

The ZenPacks.zenoss.MultiRealmIP ZenPack extends the modeling, monitoring, and event management features of Resource Manager to accommodate overlapping IP namespaces.

Note This ZenPack is not installed when Resource Manager is installed. To download it, visit the *Zenoss Support* site.

With this ZenPack, Resource Manager can prefix a realm identifier to the IP addresses of a network, enabling unified monitoring.

There are two primary use cases for using multi-realm IP management.

- A large company that manages multiple locations that have the same network spaces defined across these multiple locations and as a result have created multiple overlapping IP spaces and Resource Manager needs a way to identify each separate IP space in the system.
- Service Providers responsible for monitoring multiple customers where the customers have created independent networks and IP spaces that are unique to their location, but not unique to the Service Provider.

The essential workflow for creating and using IP Realms is that first you need to create the IP realms and then associate these realms with a collector. The associations between IP Realms and actual devices is made automatically by the device's association with the collector. All devices on a collector are associated with the realm for that collector.

Prerequisites

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

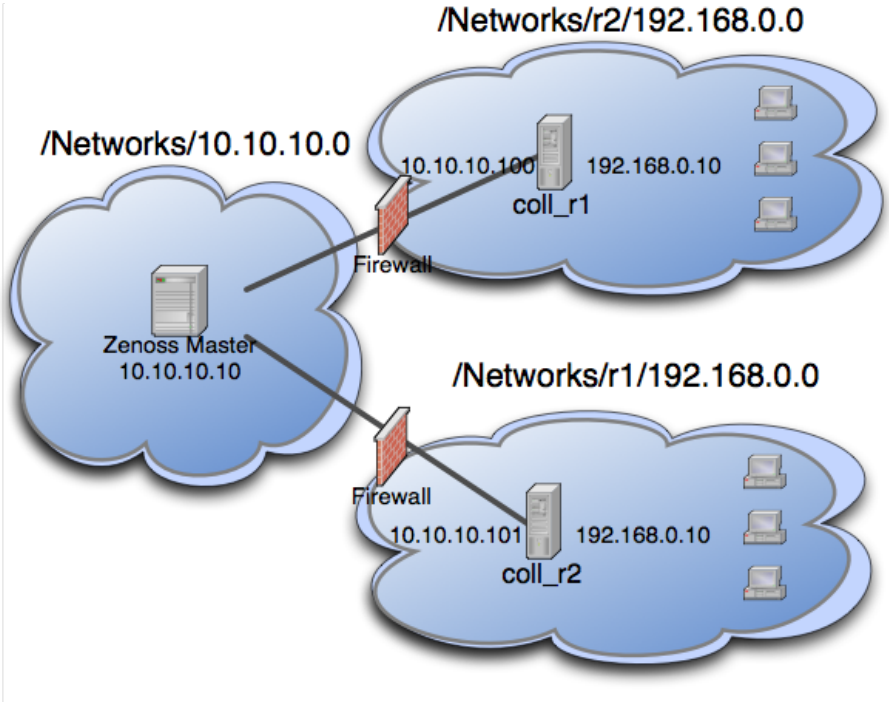
Prerequisite	Restriction
Product	Resource Manager 4.x
Required ZenPacks	ZenPacks.zenoss.DistributedCollector, ZenPacks.zenoss.MultiRealmIP

Example System

The following diagram lays out an example setup. It has a central Resource Manager server in the 10.10.10.0/24 network. The network local to the server is considered the default network within the system. The default network is treated exactly the same as a Resource Manager system without this ZenPack installed.

There are two other networks shown (`r1` and `r2`) which are behind a firewall and have the same IP space 192.168.0.0/24. Each realm has a distributed collector located within it. The collector can be accessed from the Resource Manager server using a IP translation from the firewall to map the address accessible from in front of the firewall to an address behind the firewall. Remote collectors in a multi-realm setup must be accessible from the central server using SSH.

Figure 22: Example IP Realm



System Setup

Set up Resource Manager following the example system described previously.

Note If you do not have overlapping IP space this example can be created using collectors within the same network. To create the example, add a machine multiple times once per collector, making sure to change the name of the device as it is added. The result is similar to a real realm setup.

Under multi-realm IP networks, device names *must* be unique even though the IP addresses will overlap.

On certain server configurations, if a distributed collector is configured, a "zenpack command failed" error occurs when installing this ZenPack. If you encounter this error, then run the following grant (as MySQL root):

```
grant super on *.* to 'zenoss'@'{FQDN_of_Zenoss_host}' identified by
 'zenoss';
```

where the first 'zenoss' is the user account that Resource Manager uses to access MySQL, and the second 'zenoss' is that account's password.

Adding Realms

1. Go to Infrastructure > Networks.
2. From the Add menu, select Add IP Realm. Add the realms `r1` and `r2.`

Adding Collectors to Realms

1. Add the two collectors that are installed in each realm.

   Distributed collectors now have an **IP Realm** field on their configuration screen set each collector to the appropriate realm configured above.
2. Change each collector so that it is in the correct realm.

Adding Devices to Realms

1. Now we are ready to add devices to the system. As mentioned above, adding the same device to the system twice can simulate a multi-realm setup. Add a device called `A.test` making sure that when it is added the collector is set to one of the remote collectors, and not `localhost`.
2. Now rename the device.
3. Add the device a second time using your other collector, again not `localhost`.
4. After the device is loaded, select Software and follow the network link on one of the interfaces. Notice that the network has been created underneath the realm created earlier. This configuration is at the heart of multi-realm, as networks are discovered they are created within each realm.

   Monitoring is now happening on each representation of the device from the different collectors in different overlapping realms.

As another test try searching by IP from the top-level search. Two devices will be returned -- one within each realm.

Notes

- If an event contains the unique name of a device then it is straight-forward to assign it to the proper device. If only the IP address is sent the event will be assigned by looking up the IP within the context of the realm.
- If a device is moved between realms it must be remodeled so that its IPs are placed in the proper location.
- The **Network Map** only supports the display the default realm.

(MySqlMonitor) MySQL Database Monitor 40

The ZenPacks.zenoss.MySqlMonitor ZenPack monitors MySQL database servers through the Python `twisted.enterprise.adbapi` asynchronous framework.

Note Release 4.2.5 of Resource Manager includes version 3.x of this ZenPack, which collects data through a customized modeler plugin. If you have upgraded Resource Manager from an earlier release, see *Migrating from version 2.x to 3.x* on page 148.

This ZenPack provides the following features.

- Discovery of MySQL entities
- Monitoring of MySQL Server and Database components
- Event management and monitoring for certain MySQL states
- Integration with Zenoss Service Dynamics Service Impact (Service Impact)

Discovery

The following entities are discovered through the zMySQLConnectionString property you provide. The attributes, tags and collections are updated during remodeling, which defaults to every 12 hours.

Servers

Attributes: Percentage of full table scans, Slave status, Master status, Number of databases

Tags: Name

Collections: Databases

Databases

Attributes: Number of tables, Default character set, Default collation

Tags: Name

All discovered values are valid as of the most recent modeling time.

Performance Monitoring

Server and database metrics are collected every 5 minutes by default. The Average statistic is collected, and the graphed value is per second for anything that resembles a rate.

Server metrics

Aborted clients	Aborted connects	Bytes received	Bytes sent
Com alter db	Com alter table	Com call procedure	Com check
Com commit	Com create db	Com create table	Com create user
Com delete multi	Com delete	Com drop db	Com drop table
Com drop user	Com execute sql	Com flush	Com insert select
Com insert	Com purge	Com repair	Com replace
Com rollback	Com select	Com update multi	Com update
Connections	Data size	Handler commit	Handler delete
Handler read first	Handler read key	Handler read last	Handler read next
Handler read prev	Handler read rnd next	Handler read rnd	Handler rollback
Handler savepoint	Handler update	Handler write	Index size
Key read requests	Key reads	Key writes	Max used connections
Open files	Open streams	Open tables	Select full join
Select full range join	Select range check	Select range	Select scan
Size	Threads connected	Uptime	

Database metrics

- Data size
- Index size
- Size

Event monitoring

The following table associates the MySQL components and states with the severity levels of events that are triggered in Resource Manager.

Component	Component state	Event severity
Device	Invalid zMySQLConnectionString property	Critical
	Access denied for user with credentials provided in zMySQLConnectionString property	Critical
Service	Slave error	Error
	MySQL server is not accessible	Error
	InnoDB deadlock	Warning
	A database was dropped	Info
Database	A table was dropped	Warning
	A table was added	Info

Integration with Service Impact

This ZenPack includes custom state providers for services running on MySQL hosts. Custom state providers enable specialized options for defining state triggers in Zenoss Service Dynamics Service Impact (Service Impact).

The following relationships are automatically included in any Service Impact services that contain one or more of the explicitly mentioned components.

- Server failure affects related device
- Database failure affects related server

The preceding relationships follow the default policy, in which a node inherits the worst state of the ancestor nodes to which it is related. For example, a server failure implies that all related databases are also failed.

Prerequisites

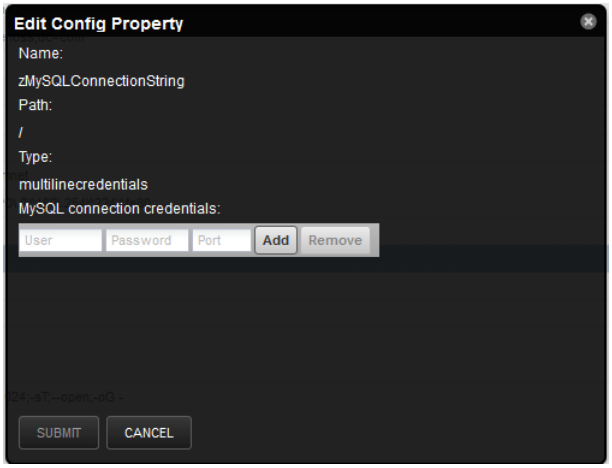
Prerequisite	Restriction
Product	Resource Manager 4.x
Required ZenPacks	ZenPacks.zenoss.MySqlMonitor
	ZenPacks.zenoss.PythonCollector, version 1.1 or higher
Link for remote Zenoss DataStore	To monitor Zenoss DataStore with this ZenPack, see *Enabling Zenoss DataStore monitoring* on page 148.

Adding a MySQL Server instance

To monitor a MySQL Server instance through a user account other than `root`, grant SELECT privileges to the account, before adding the instance to Resource Manager. For more information, refer to *MySQL GRANT syntax*.

Follow these steps to monitor a MySQL Server instance with Resource Manager.

1. Log in to Resource Manager browser interface as a user with ZenManager or Manager privileges.
2. Navigate to the **INFRASTRUCTURE > Devices** page.
3. In the left column, select the device class of the MySQL Server host.
4. In the device list, click the host's entry, to display its details page.
5. Configure the zMySQLConnectionString property.
   a. In the left column, select **Configuration Properties**.
   b. In the properties list, double-click the zMySQLConnectionString property.



c In the **Edit Config Property** dialog, enter a valid MySQL Server user account and password, and the port on which the MySQL Server instance listens for connection requests.

Note Resource Manager version 4.1.x does not support the customized dialog for MySQL Server credentials. Create a JSON list instead.

d Click **SUBMIT**.

6 Add the MySQL plugin.

a In the left column, select **Modeler Plugins**.

b In the list of available plugins, select **MySQL Collector**.

c Click the right arrow button.

d At the bottom of the page, click **Save**.

Enabling Zenoss DataStore monitoring

Follow these steps to enable this ZenPack to monitor a Zenoss DataStore instance that is not located on the Resource Manager master host.

1 Log in to the remote Zenoss DataStore host as `zenoss`.

2 Create a symbolic link to the Zenoss DataStore socket file.

```
ln -s /var/lib/zends/zends.sock /tmp/mysql.sock
```

Migrating from version 2.x to 3.x

Perform this procedure if one of the following statements is true.

- You have upgraded Resource Manager to release 4.2.5, or to a more recent release.
- You have manually updated this ZenPack from version 2.x to version 3.x, or to a more recent version.

Note Version 3.x uses a customized modeler plugin to collect data, and data collected through version 2.x of this ZenPack is incompatible. The version 2.x data is preserved. New RRD files are created for the new data.

1 Log in to Resource Manager browser interface as a user with ZenManager or Manager privileges.

2 Navigate to the **INFRASTRUCTURE > Devices** page.

3 From the device list, select the MySQL Server host.

4 From the action menu at the bottom of the left column, select **Bind Templates**.

5 In the **Bind Templates** dialog, select **MySQL (/Server)** from the **Selected** column.

6 Click the left arrow button to move the template to the **Available** column.

7 Click **SAVE**.

8 From the action menu at the bottom of the left column, select **Model Device...**.

Installed items

This ZenPack adds the following items to Resource Manager.

Modeler Plugins

MySQLCollector

Monitoring Templates

MySQLServer(in /Server)

MySQLDatabase (in /Server)

Component Types

MySQLServer (on related device)

MySQLDatabase (on MySQLServer)

41 (NetScreenMonitor) NetScreen Monitor

The ZenPacks.zenoss.NetScreenMonitor ZenPack monitors devices from NetScreen Technologies.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.NetScreenMonitor

Configuring NetScreen Devices to Allow SNMP Queries

Configure the NetScreen device to allow SNMP queries from the Resource Manager server, and send SNMP v1 or SNMP v2 traps to the Resource Manager server.

Configuring Resource Manager

All NetScreen devices must exist under the `/Devices/Network/NetScreen` device class.

1 Navigate to the device or device class in the Resource Manager interface.

- If applying changes to a device class:
  1 Select the class in the devices hierarchy.
  2 Click **Details**.
  3 Select Configuration Properties.
- If applying changes to a device:
  1 Click the device in the device list.
  2 Select Configuration Properties.

2 Edit the appropriate configuration properties for the device or devices.

Name	Description
zSnmpCommunity	Consult with your network administrators to determine the SNMP community permitted.

Name	Description
zSnmpMonitorIgnore	This should be set to `False`
zSnmpPort	The default port is 161.
zSnmpVer	This should be set to `v2c`

3 Click **Save** to save your changes. You will now be able to start collecting the NetScreen device metrics from this device.

4 Navigate to Graphs and you should see some placeholders for graphs. After approximately fifteen minutes you should see the graphs start to become populated with information.

Daemons

Type	Name
Modeler	zenmodeler
Performance Collector	zenperfsnmp

42 (NNTPMonitor) NNTP Monitor

The ZenPacks.zenoss.NNTPMonitor ZenPack monitors the response time of Network News Transfer Protocol (NNTP) servers.

The response time unit of measurement is milliseconds.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.NNTPMonitor

Enable Monitoring

To enable monitoring for a device:

1. Select Infrastructure from the navigation bar.
2. Click the device name in the device list.

   The device overview page appears.
3. Expand Monitoring Templates, and then select Device from the left panel.
4. Select Bind Templates from the Action menu.

   The Bind Templates dialog appears.
5. Add the NNTPMonitor template to the list of selected templates, and then click **Submit**.

   The NNTPMonitor template is added to the list of monitoring templates.
6. Select the template and change options as needed.
7. Validate your configuration by running `zencommand` and observing that the `check_nntp` or `check_nntps` command correctly connects to your NNTP server:

```
zencommand run -v10 -d yourdevicenamehere
```

Daemons

Type	Name
Performance Collector	zencommand

(NortelMonitor) Nortel Monitor 43

The ZenPacks.zenoss.NortelMonitor ZenPack monitors devices from Nortel Networks.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.NortelMonitor

Configuring Nortel Devices to Allow SNMP Queries

Configure the Nortel device to allow SNMP queries from the Resource Manager server, and send SNMP v1 or SNMP v2 traps to the Resource Manager server.

Configuring Resource Manager

All Nortel devices must exist under the `/Devices/Network/Nortel` device class.

1. Navigate to the device or device class in the Resource Manager interface.
   - If applying changes to a device class:
     1. Select the class in the devices hierarchy.
     2. Click **Details**.
     3. Select Configuration Properties.
   - If applying changes to a device:
     1. Click the device in the device list.
     2. Select Configuration Properties.
2. Edit the appropriate configuration properties for the device or devices.

Table 41: Nortel Configuration Properties

Name	Description
zSnmpCommunity	Consult with your network administrators to determine the SNMP community permitted.
zSnmpMonitorIgnore	This should be set to `False`
zSnmpPort	The default port is 161.
zSnmpVer	This should be set to `v2c`

3 Click **Save** to save your changes. You will now be able to start collecting the Nortel device metrics from this device.

4 Navigate to Graphs and you should see some placeholders for graphs. After approximately fifteen minutes you should see the graphs start to become populated with information.

Daemons

Type	Name
Modeler	zenmodeler
Performance Collector	zenperfsnmp

44 (NtpMonitor) NTP Monitor

The ZenPacks.zenoss.NtpMonitor ZenPack monitors the difference between the system time a server is using and the time a Network Time Protocol (NTP) server is reporting.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.NtpMonitor

Enable Monitoring

The NTPMonitor template must be bound to the device class or device you want to monitor.

1. Select Infrastructure from the navigation bar.
2. Click the device name in the device list.

   The device overview page appears.
3. Expand Monitoring Templates, and then select Device from the left panel.
4. Select Bind Templates from the Action menu.

   The Bind Templates dialog appears.
5. Add the NTPMonitor template to the list of selected templates, and then click **Submit**.

   The NTPMonitor template is added to the list of monitoring templates. You can now start collecting the NTP server metrics from this device.

Daemons

Type	Name
Performance Collector	zencommand

45 (RANCIDIntegrator) RANCID Integration

The ZenPacks.zenoss.RANCIDIntegrator ZenPack enables integration between the RANCID configuration management tool and Resource Manager.

The integration points are:

- Resource Manager will build the `router.db` file for *RANCID*. This allows for the centralization of administration activities and reduces the duplication of effort normally required to maintain the two tools.
- Implementation of this feature is as easy as adding a `cron` job to execute `$ZENHOME/bin/zenrancid` to update the `router.db` file.
- Resource Manager will automatically run RANCID's `rancid-runm` tool on a single device in response to a `ciscoConfigManEvent` SNMP trap being sent from the device to Resource Manager. Cisco devices will send this trap whenever their configuration is changed. This allows for real-time capturing of router configuration changes in your CVS repository.

Note The RANCID integrator is dependent on a connection to the Zope server, hence it can run only on the Resource Manager master and as such works only with managed resources on the master.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.RANCIDIntegrator

Configure Cisco Devices to Send Traps

To implement this feature you must configure your Cisco devices to send their SNMP traps to the Resource Manager server.

Link from Cisco device status pages to the most recent configuration stored in your CVS repository via *viewvc*.

Configure RANCID Update Information in Resource Manager

1. From Infrastructure >Devices, click the device in the device list.
2. Select Configuration Properties in the left panel.
3. Edit the appropriate configuration properties for the device.

Table 42: RANCID Configuration Properties

Name	Description
zRancidRoot	File system directory where RANCID is installed. It may be NFS mounted from the RANCID server. Default is `/opt/rancid`
zRancidUrl	Base URL to `viewvc`
zRancidGroup	RANCID group attribute. Controls what `router.db` file the device is written to. Can be set at the device class or device level. Default is `router` on the `/Network/Router/Cisco` class
zRancidType	RANCID type attribute. Controls what device type is written to the `router.db` file. Can be set at the device class or device level. Default is `cisco` on the `/Network/Router/Cisco`

4 Click **Save** to save your changes.

46 RPCMonitor (RPC Monitor)

The ZenPacks.zenoss.RPCMonitor ZenPack monitors applications that use Open Network Computing (ONC) Remote Procedure Call (RPC).

Note This ZenPack is not installed when Resource Manager is installed. To download it, visit the *Zenoss Support* site.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x
Required ZenPacks	ZenPacks.zenoss.RPCMonitor

Enable Monitoring

The RPCMonitor template must be bound to the device class or device you want to monitor. Follow these steps to enable monitoring:

1. Select Infrastructure from the navigation bar.
2. Click the device name in the device list.

   The device overview page appears.
3. Select Configuration Properties from the left panel.
4. Set the appropriate RPC command to test in the zRPCCommand configuration property (for example, nfs or ypserv).
5. Click **Save**.
6. Expand Monitoring Templates, and then select Device from the left panel.
7. Select Bind Templates from the Action menu.

   The Bind Templates dialog appears.
8. Add the RPCServer template to the list of selected templates, and then click **Submit**.

   The RPCServer template is added to the lists of monitoring templates. You can now collect the RPCServer server metrics from the device.

Daemons

Type	Name
Performance Collector	`zencommand`

47 Splunk (Splunk)

The ZenPacks.zenoss.Splunk ZenPack facilitates unified monitoring by invoking customized searches of Splunk databases.

Note This ZenPack is not installed when Resource Manager is installed. To download it, visit the *Zenoss Support* site.

Splunk is a search engine for IT data. It lets you search and analyze all the data your IT infrastructure generates from a single location in real time. More information on Splunk can be found online at *http://www.splunk.com/*.

This ZenPack allows you to monitor the results of a Splunk search. The total count returned by a search can be recorded, thresholded and graphed as well as additional tabular data contained within the results of more advanced searches that make use of Splunk's top filter. The value of monitoring Splunk searches is that it adds an easy and flexible way to monitor log data at aggregate level instead of on a log-by-log basis.

Prerequisites

Note Resource Manager does not support the free version of Splunk.

Prerequisite	Restriction
Product	Resource Manager 4.x
Required ZenPacks	ZenPacks.zenoss.Splunk
Third Party Software	Splunk Version 3 or 4

Splunk Data Source Type

This ZenPack adds the Splunk data source type to Resource Manager. This data source can be used to monitor the results of Splunk searches.

The Splunk data source type has the following fields in common with many other Resource Manager data source types:

- **Name**: The name given to your data source.
- **Enabled**: This data source will only be polled if enabled is set to true.

In the event that the Splunk search fails to execute successfully an event will be generated. The following fields control key fields in the generated event. It is important to note that these fields only apply when the Splunk search fails to execute, and not when a threshold on the data point is breached.

- **Component**
- **Event Class**
- **Event Key**
- **Severity**

The following fields are specific to Splunk type data sources.

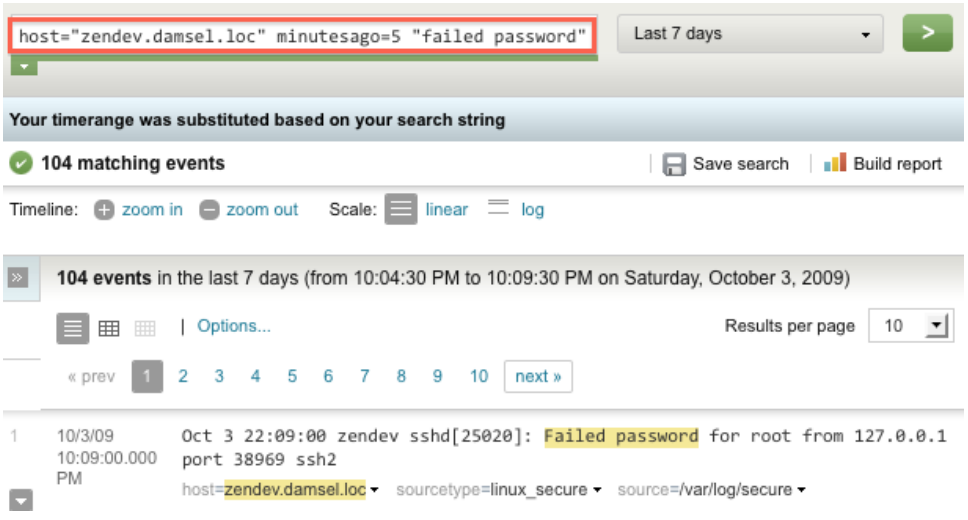
- **Splunk Server**: Hostname or IP address of your Splunk server. If left blank the SPLUNK_SERVER environment variable will be used.
- **Splunk Port**: Port that the splunkd daemon is listening on. Default is 8089. If left blank the SPLUNK_PORT environment variable will be used.
- **Splunk Username**: Splunk username. Default is admin. If left blank the SPLUNK_USERNAME environment variable will be used.
- **Splunk Password**: Splunk password. Default is changeme. If left blank the SPLUNK_PASSWORD environment variable will be used.
- **Search**: Search string exactly as it would be typed into the Splunk search engine. Be careful to use full quotes and not apostrophes where necessary.

Monitoring Results of a Simple Search

The easiest way to get started monitoring your Splunk searches is with a simple search. The following steps will illustrate a simple way to build dynamic Splunk search monitoring.

This example demonstrates how to detect brute-force password cracking attempts on all Linux servers.

1 Build a search in Splunk to verify that you're getting the expected data. This example shows a query of `host="zendev.damsel.loc" minutesago=5 "failed password"`.



Note Using a time specifier such as `minutesago=5` within your search can be a useful trick when it comes to monitoring searches from Resource Manager. We will have Resource Manager automatically replace `zendev.damsel.loc` with the appropriate hostname using a `${here/id}` TALES expression.

2 Create a Resource Manager monitoring template for monitoring this Splunk search.

a From Advanced > Monitoring Templates, click Add to add a monitoring template.

The Add Template dialog appears.

b Enter SplunkLinux in the Name field and select Linux in /Service/Linux for Template Path, and then click **Submit**.

c Select the newly created template.

d Add a Splunk data source to capture the count of failed passwords.

a In the Data Sources area, click Add to add a data source.

b In the Add Data Source dialog, set the Name to `failedPassword` and the **Type** to Splunk, and then click **OK**.

c Double-click the data source to configure it as follows, and then click **Save**.

- **Splunk Server**: *Hostname or IP of your Splunk server*
- **Splunk Port**: `8089`
- **Splunk Username**: *Splunk username* (default is admin)
- **Splunk Password**: *Splunk password* (default is changeme)
- **Search**: `host="${here/id}" minutesago=5 "failed password"`

d Add the *count* data point to the *failedPassword* data source.

a Select Add Data Point from the Data Sources Action menu.

b Set the **Name** to `count` and click **OK**.

e Add a threshold of how many failed passwords constitutes an attack.

a In the Thresholds area, click Add to add a threshold.

b Set the Name to `password attack` and **Type** to MinMaxThreshold, and then click Add..\

c Select failedPassword_count from **Data Points**.

d Set the **Max Value** to `10`.

e Set the **Event Class** to /Security/Login/BadPass.

f Click **Save**.

f Add a graph to visualize failed passwords per 5 minutes.

a In the Graph Definitions area, click Add to add a graph.

b Set the Name to `Splunk - Failed Passwords`, and then click **Submit**.

c Double-click the newly created graph to edit it.

d Set the **Units** to `failed/5min`.

e Set the **Min** Y to `0`.

f Select Manage Graph Points from the Action menu in the Graph Definitions area.

The Manage Graph Points dialog appears.

g Select Data Point from the Add menu.

The Add Data Point dialog appears.

h Select failedPassword_count from **Data Point**, and then click **Submit**.

i Click into the new count graph point.

j Set the **RPN** to `300,*` to adjust from failed/sec to failed/5min.

k Set the **Format** to `%6.1lf`.

l Set the **Legend** to `Count`.

m Click **Save**.

g Bind the SplunkLinux template to the `/Server/Linux` device class.

a From Infrastructure > Devices, navigate to the `/Server/Linux` device class.

b Click Details.

c Select Bind Templates from the Action menu.

d Move the SplunkLinux template from the Available area to the Selected area, and then click **Save**.

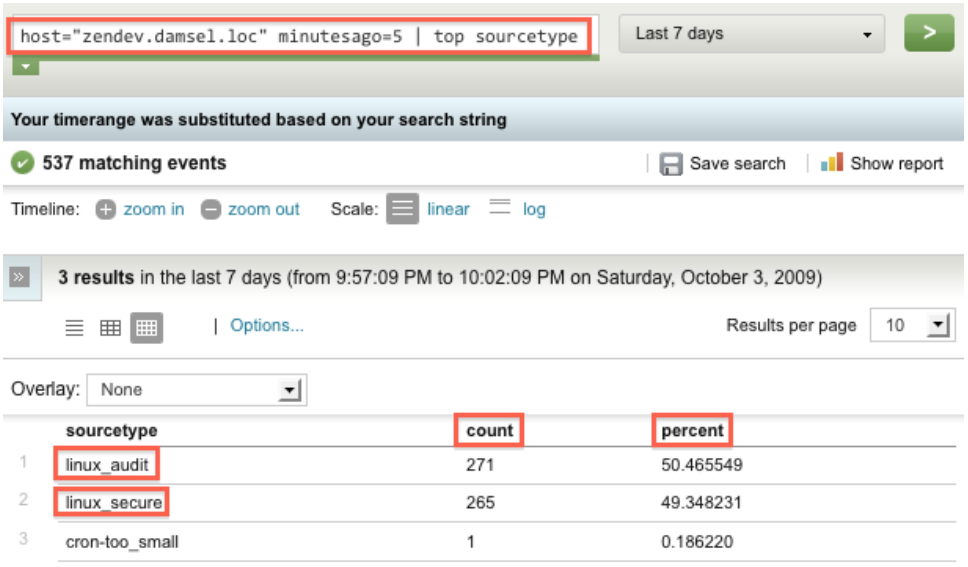
Now you will have a Failed Passwords graph on all of your Linux servers that visualizes how many failed password attempts have occurred over the last 5 minutes. You will also get a warning severity event anytime more than 10 failed password attempts are made within a 5 minute period.

Monitoring Results of a Top Search

Monitoring additional data points within a top search builds on monitoring a simple search. You can extra numeric data from the tabular results returned from a top search using the following steps.

This example demonstrates how you can monitor the logs by source type for all Linux devices.

1 Build a search in Splunk to verify that you're getting the expected data. This example shows a query of `host="zendev.damsel.loc" minutesago=5 | top sourcetype`.



Note Take special note of the names in the sourcetype column and the names of the count and percent columns. These will be used to construct the names of the datapoints within our Splunk data source.

2 Setup a Resource Manager monitoring template just as described in the simple search example.

3 Add a Splunk type data source named `sourcetype` to the template with the following settings.

- **Splunk Server**: *Hostname or IP of your Splunk server*
- **Splunk Port**: `8089`
- **Splunk Username**: *Splunk username* (default is admin)
- **Splunk Password**: *Splunk password* (default is changeme)
- **Search**: `host="${here/id}" minutesago=5 | top sourcetype`

4 Add data points to the sourcetype data source with the following names. These names come from concatenating the data in the first column of each row with the name of the column name with the target numeric data.

- `linux_audit_count`
- `linux_audit_percent`
- `linux_secure_count`
- `linux_secure_percent`

5 Create a graph that will show these results within Resource Manager in a useful way.

a Add a graph from the Graph Definitions area of the monitoring template.
b Set the **ID** to `Splunk - Logs by Source Type` then click **Submit**.
c Set the **Units** to `percent`.
d Set the **Min Y** to `0`.
e Set the **Max Y** to `100`.
f Click **Save**.
g Select Manage Graph Points from the Action menu in the Graph Definitions area.

The Manage Graph Points dialog appears.

h Select Data Point from the Add menu.

The Add Data Point dialog appears.

i Use SHIFT-click or CTRL-click to select the following data points from the list then click **Submit**.

- sourcetype_linux_audit_percent
- sourcetype_linux_secure_percent

j Click into each of the graph points you just added to the graph and set the following properties.

- **Line Type**: Area
- **Stacked**: True
- **Format**: `%5.1lf%%`
- **Legend**: `Audit` or `Secure` respectively.

6 Bind the monitoring template to the `/Server/Linux` device class just as in the simple search example.

You will now have a graph for all Linux devices that shows what percentage of logs are coming from the audit and secure logs respectively. This ability to track multiple results from a single Splunk search has many other possible uses. Experiment with the top filter in Splunk to see what other useful data you could extract.

Daemons

Type	Name
Performance Collector	`zencommand`

48 (StorageBase) Storage Base

The ZenPacks.zenoss.StorageBase ZenPack contains base classes and reports for ZenPacks that use the base classes.

Reports include:

- **Licenses** - Shows the storage devices and installed licenses.
- **Clients** - Shows the devices that use the storage devices.
- **Disk Firmware** - After selecting a storage device, displays disk firmware information.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 3.0 or higher
Required ZenPacks	ZenPacks.zenoss.StorageBase, ZenPacks.zenoss.DynamicView

49 (SugarCRMMonitor) SugarCRM Monitor

The ZenPacks.zenoss.SugarCRMMonitor ZenPack monitors SugarCRM services.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.SugarCRMMonitor

Configuring Resource Manager

All SugarCRM devices must exist under the `/Devices/Web/SugarCRM` device class.

1. Navigate to the device or device class under the `/Devices/Server/Tomcat` device class in the Resource Manager interface.
   - If applying changes to a device class:
     1. Select the class in the devices hierarchy.
     2. Click **Details**.
     3. Select Configuration Properties.
   - If applying changes to a device:
     1. Click the device in the device list.
     2. Select Configuration Properties.
2. Edit the appropriate configuration properties for the device or devices.

Table 43: SugarCRM Configuration Properties

Name	Description
zSugarCRMBase	
zSugarCRMPassword	Password for the *zSugarCRMUsername* user.

Name	Description
zSugarCRMTestAccount	
zSugarCRMUsername	Username allowed to log into the Sugar CRM server.

3 Click **Save** to save your changes.
4 From the left panel, select Device under Monitoring Templates.
5 Select Bind Templates from the Action menu.

The Bind Templates dialog appears.
6 Move the SugarCRM template from the Available list to the Selected list.
7 Click **Save**.

The **SugarCRM** template should now be displayed under the **Monitoring Templates for** *Device*. You will now be able to start collecting the Sugar CRM metrics from this device.
8 Navigate to Graphs and you should see some placeholders for graphs. After approximately fifteen minutes you should see the graphs start to become populated with information.

Daemons

Type	Name
Performance Collector	zencommand

(TomcatMonitor) Apache Tomcat 50

The ZenPacks.zenoss.TomcatMonitor ZenPack monitors Apache Tomcat servers.

Tomcat is a Web application container that conforms to many parts of the J2EE Specification.

This ZenPack focuses on the metrics that Tomcat updates in its internal MBean container that is accessible via the remote JMX API. These metrics focus on attributes that relate to the servicing of web pages and primarily include thread pool size, CPU use, available file descriptors, JSP and servlet counts, and request counts.

This ZenPack places much emphasis on monitoring thread status because every web request is serviced in a separate thread. Each thread requires file descriptors to be maintained, and thus those are monitored as well. The amount of CPU time spent servicing each thread is also captured and reported.

This ZenPack also reports on the number of times JSPs and Servlets are reloaded. This metric can be useful in highly dynamic sites where JSPs or Servlets change on the fly and need to be reloaded periodically. Monitoring of this metric can lead to the identification of small "Reloading Storms" before they cause production outages.

The amount of time Tomcat spends servicing a request is also recorded. This extremely high level metric can provide insight into downstream systems that are not monitored. If all the Tomcat resources are within normal tolerances but processing time suddenly spikes it can be an indication that a back-end service (such as a database or another web service) is misbehaving.

The following metrics can be collected and graphed:

- Tomcat cache (accesses vs hits)
- Daemon and User thread count
- Overall CPU time
- Global Request Traffic: bytes sent/received
- Global Request Traffic: request count and error count
- Global Request processing time
- JSP/Servlet reload time
- Servlet class loading and processing time
- Servlet request and error count

Note The more extensive JBoss Application Server uses Tomcat as a Web Application engine to manage web applications deployed inside enterprise applications within JBoss. As a result, this ZenPack can be used to monitor Tomcat MBeans that are active within JBoss.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.ZenJMX, ZenPacks.zenoss.TomcatMonitor

Configuring Tomcat to Allow JMX Queries

Before running the Tomcat `bin/start.sh` script, run the following to allow unsecured queries against the Tomcat server:

```
JAVA_OPTS="-Dcom.sun.management.jmxremote.port=12346"
JAVA_OPTS="${JAVA_OPTS} -
Dcom.sun.management.jmxremote.authenticate=false"
JAVA_OPTS="${JAVA_OPTS} -Dcom.sun.management.jmxremote.ssl=false"
export JAVA_OPTS
```

The same JAVA_OPTS approach can be used to enable remote access to Tomcat MBeans. Set the JAVA_OPTS variable as illustrated above and then execute the `./catalina.sh start` command in the `${TOMCAT_HOME}/bin` directory.

Note Tomcat 6.0.14's `catalina.sh` does not process the `stop` command properly when the JAVA_OPTS variable is set. We recommend using two separate shell scripts when troubleshooting JMX problems in Tomcat: one for starting Tomcat (with the JAVA_OPTS variable set) and a different one for stopping Tomcat (where the JAVA_OPTS variable is not set).

If you add the above lines to the to `bin/setenv.sh` (as seems to be the logical thing to do in `catalina.sh` to get the environment variables set up), the `bin/shutdown.sh` script will get those same environment variables. This will cause the `shutdown.sh` script to attempt to bind to the ports, fail, and then not stop Apache Tomcat.

Configuring Resource Manager

All Apache Tomcat services must have a device entry under the `/Devices/Server/Tomcat` device class.

Note The `zenjmx` daemon must be configured and running. See for more information about configuring the `zenjmx` daemon with the Sun JRE tools.

1. Navigate to the device or device class under the `/Devices/Server/Tomcat` device class in the Resource Manager interface.
   - If applying changes to a device class:
     1. Select the class in the devices hierarchy.
     2. Click **Details**.
     3. Select Configuration Properties.
   - If applying changes to a device:
     1. Click the device in the device list.

2 Select Configuration Properties.

2 Edit the appropriate configuration properties for the device or devices.

Table 44: Tomcat Configuration Properties

Name	Description
zTomcatJ2EEApplicationName	Used to construct MBean names for a specific application deployed on Tomcat, typically used for JSP and Servlet statistics.
zTomcatJ2EEServerName	Used to construct MBean names for a specific application deployed on Tomcat, typically used for JSP and Servlet statistics.
zTomcatJmxManagementAuthenticate	This configuration property is deprecated.
zTomcatJmxManagementPassword	JMX password.
zTomcatJmxManagementPort	The port number used to gather JMX information.
zTomcatJmxManagementUsername	JMX username for authentication.
zTomcatListenHost	The hostname on which Tomcat is listening for web requests. This is used to construct MBean names.
zTomcatListenPort	The Tomcat connector, which is a port and protocol (http, jk...) that Tomcat is listening on. This is used to construct MBean names that monitor bytes, error and requests on that connector.
zTomcatServletName	Specific Servlet name to monitor.
zTomcatServletUri	URI of Servlet to monitor.
zTomcatWebAppUri	URI path for a Tomcat web application. Used to construct MBean names.

3 Click **Save** to save your changes.

You will now be able to start collecting the Tomcat server metrics from this device.

4 Navigate to Graphs and you should see some placeholders for performance graphs. After approximately fifteen minutes you should see the graphs start to become populated with information.

Note The out-of-the-box TomcatMonitor data source configuration has been defined at the macro level, but can be configured to operate on a more granular basis. For example, the Servlet Reload Count applies to all servlets in all web applications but it could be narrowed to be Servlet /submitOrder in web application "production server".

Change the Amount of Data Collected and Graphed

1 Navigate to the device or device class under the /Devices/Server/Tomcat device class in the Resource Manager interface.

2 From the left panel, select Monitoring Templates.

3 From the Action menu, select Bind Templates.
4 Move one or more templates to Selected, and then click **Save**.

Table 45: Tomcat Templates

Name	Description
`Tomcat Cache`	Cache information about a specific Web application deployed.
`Tomcat Core`	Core information about any Tomcat server: memory usage, threads, uptime, etc.
`Tomcat Global Request Processor`	Connection information over a Tomcat connector: bytes, errors, requests.
`Tomcat JSPS`	Metrics about a specific JSP page.
`Tomcat Servlet`	Metrics about a specific Servlet.
`Tomcat Thread Pool`	Threadpool metrics measured per Tomcat connector.
`Tomcat Web Module`	Processing time metrics for a Web module.

5 Click the **OK** button to save your changes.

Viewing Raw Data

For more information about investigating raw data returned from Apache Tomcat, see *Using JConsole to Query a JMX Agent* on page 220.

Daemons

Type	Name
Performance Collector	zenjmx

(vCloud) VMware vCloud 51

The ZenPacks.zenoss.vCloud ZenPack monitors virtual infrastructure services that are managed by VMware vCloud Suite platforms.

VMware vCloud acts as a cloud layer on top of one or more vSphere virtual infrastructures. It allows for easy deployment of public or private clouds with required concepts, such as a self-service portal with built-in multi-tenancy. vCloud enables you to allocate your vSphere resources as desired to provide abstracted compute (CPU and memory) and storage resources to internal or external customers.

You can find more information about cloud computing and vCloud at the VMware site:

http://www.vmware.com

This ZenPack uses VMware's Cloud Director native management API to extend in-depth availability, performance, and event monitoring into the vCloud platform. If you provide the vCloud service, you can use the vCloud administrator perspective to gain a complete view of the entire cloud architecture. Or, as a vCloud consumer, you can use the user perspective to obtain organization-specific information.

This ZenPack adds the **zenvcloud** daemon to Resource Manager.

Prerequisites

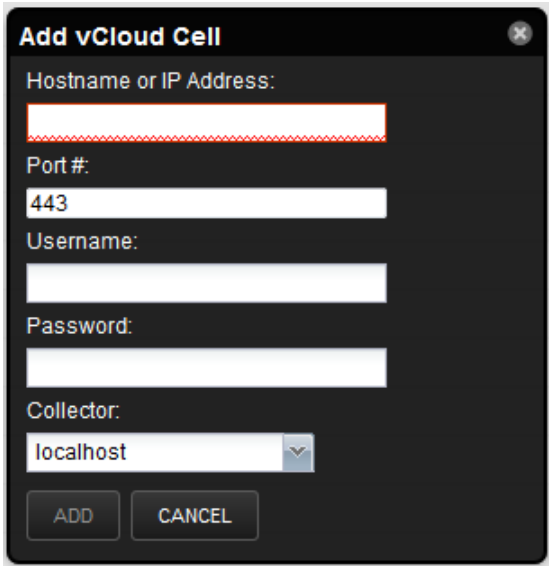
Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 3.0.x or higher
Required ZenPacks	ZenPacks.zenoss.vCloud
Application	VMware vCloud, version 1.0, 1.5, or 5.1

Adding a vCloud cell

This ZenPack discovers and monitors vCloud resources through vCloud cells, also known as the cloud director or self-service portal.

Follow these steps to add a vCloud cell through the Resource Manager browser interface.

1. Log in to Resource Manager user interface as a user with ZenManager or Manager privileges.
2. Navigate to the **INFRASTRUCTURE** page.
3. From the the **Add Device** menu, select **Add vCloud Cell...**.



4 Complete the dialog with the information for the vCloud cell to add.

Hostname or IP Address	The hostname or IP address of the vCloud cell to monitor. The cell must be accessible, and its hostname resolvable, from the Resource Manager collector host.
Port #	The HTTPS port of the vCloud cell to monitor. The default is 443.
Username	A user account name the vCloud cell accepts, in the form `username@organization`. For example, `administrator@acme`.
Password	The password of the user account the vCloud cell accepts.
Collector	The hostname of the Resource Manager collector for this vCloud cell. The default is `localhost` (the Resource Manager master host).

5 Click **ADD**.

Monitoring

The `zenvcloud` daemon uses the VMware Cloud Director API to perform monitoring. Once a cell is added to Resource Manager, monitoring begins automatically.

Resource Manager collects the following metrics directly from a vCloud cell for each vDC.

- CPU Limit, Allocated and Used
- Memory Limit, Allocated and Used
- Storage Limit, Allocated and Used

If you supply administrator credentials, Resource Manager collects the following metrics directly from a vCloud cell for each Provider vDC:

- CPU Capacity, Allocation and Free
- Memory Capacity, Allocation and Free
- Storage Capacity, Allocation and Free

(VMwareESXMonitor) VMware ESX SNMP 52

The ZenPacks.zenoss.VMwareESXMonitor ZenPack monitors VMware ESX hosts and their guests, using SNMP.

Note This ZenPack is deprecated; see *(vSphere) VMware vSphere* on page 177.

With this ZenPack, the `zenmodeler` daemon can discover guests running on ESX hosts, and provide screens and templates for collecting and displaying resources allocated to guests.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x
Required ZenPacks	ZenPacks.zenoss.VMwareESXMonitor, ZenPacks.zenoss.ZenossVirtualHostMonitor

Monitoring VMware ESX Servers

To monitor VMware ESX servers:

1. Make sure you have SNMP connectivity to your ESX 3 servers.
2. Create your ESX services using the /Servers/Virtual Hosts/ESX device class.

   Note If you have already modeled these servers, then remove and recreate them under the ESX device class. Do not move them.

3. Select the Guest menu and ensure that the guest hosts were found when the devices were added.
4. Using the VMware vSphere client, add Resource Manager to the list of destinations for SNMP traps. (See Administration > vCenterServerSettings > SNMP.) For information about configuring traps for a stand-alone ESX 3 server, see "About SNMP and VMware Infrastructure" at:

   http://www.vmware.com/pdf/vi3_35/esx_3/r35u2/vi3_35_25_u2_admin_guide.pdf

Notes:

- There is a link to the VMware Web interface on each ESX server Status page.

- If the name of the Guest under ESX is the same as the name of a device being monitored directly by Resource Manager, a link is provided to take you directly to that device from the Guest list.

Enabling SNMP Subagents

ESX servers (Version 4.x and higher) contain an SNMP subagent from VMware. This subagent provides all information related to VMware (such as virtual machines and their status). By default, the subagent is disabled.

The VMware SNMP subagent does not provide information about the ESX server itself (such as processes, memory, CPU, or performance data).

Note The VMware SNMP subagent cannot share port 161. If any other agent is using that port (usually the NET-SNMP agent), the subagent cannot start.

To fully monitor the ESX machine on your Resource Manager server, you must enable both SNMP agents (NET-SNMP and the VMware subagent). Follow these steps to enable both agents using an SNMP proxy:

1 Stop the snmpd service through the service console (via SSH) on the ESX host:

```
service snmpd stop
```

2 Add a proxy line to the `/etc/snmp/snmpd.conf` file:

```
proxy -v 1 -c public udp:127.0.0.1:171 .1.3.6.1.4.1.6876
```

This line will use the snmpd service to access the VMware MIB on the subagent running at port 171.

3 Using the VMware vSphere CLI (command line interface), bind the VMware SNMP agent to port 171, and then enable the subagent by using these commands:

```
vicfg-snmp.pl --server <hostname|IP address> --username <username> --
password <password> -c \
 public --port 171
vicfg-snmp.pl --server <hostname|IP address> --username <username> --
password <password> -E
```

4 Via SSH, go back to the ESX host. Restart the mgmt-vmware service (hostd) and the snmp service. On the ESX host from the service, enter:

```
service mgmt-vmware restart
service snmpd restart
```

Daemons

Type	Name
Modeler	`zenmodeler`
Performance Collector	`zenperfsnmp`

53 (vSphere) VMware vSphere

The ZenPacks.zenoss.vSphere ZenPack monitors VMware vSphere systems and services through a vCenter server, using the vSphere API.

Note This ZenPack replaces the following, deprecated ZenPacks.

- ZenPacks.zenoss.EsxTop
- ZenPacks.zenoss.VMwareESXMonitor
- ZenPacks.zenoss.ZenVMware

Installing this ZenPack does not automatically migrate monitoring of VMware vSphere resources, or disable monitoring through the deprecated ZenPacks. For more information about transitioning to this ZenPack, see *Transitioning to this ZenPack* on page 182.

Discovery

The following vCenter components are automatically discovered, and the properties and relationships are continually maintained through a persistent subscription to the vSphere updates API.

Component	Properties	Relationships
Datacenters	None	Clusters and Standalone Compute Resources, Resource Pools, vApps, VMs, Hosts, Networks, dvSwitches, dvPortgroups, Datastores
Clusters and Standalone Compute Resources	Total Hosts, Effective Hosts, CPU Cores, CPU Threads, Total CPU, Effective CPU, Total Memory, Overall Status	Datacenter, Host(s), Resource Pools
Resource Pools	CPU Limit, CPU Reservation, Memory Limit, Memory Reservation, Overall Status	Datacenter, Cluster or Standalone Compute Resource, Parent Resource Pool, Child Resource Pools, vApps, VMs
vApps	CPU Limit, CPU Reservation, Memory Limit, Memory Reservation, Overall Status	Datacenter, Cluster or Standalone Compute Resource, Resource Pool, VMs
VMs	OS Type, Guest Name, Memory, CPUs, Template?, Connection State, Power State, Overall Status	Datacenter, Resource Pool or vApp, vNICs, Host, Datastores

Component	Properties	Relationships
vNICs	MAC Address	VM, Network or dvPortgroup
Hosts	Hostname, Total Memory, In Maintenance Mode?, Connection State, Power State, Overall Status	Datacenter, Cluster or Standalone Compute Resource, VMs, Host NICs, Datastores, LUNs
Host NICs	MAC Address	Host
Networks	Accessible?, IP Pool Name	Datacenter, Attached vNICs
dvSwitches	None	Datacenter, dvPortgroups
dvPortgroups	Accessible?, IP Pool Name, Key UUID	Datacenter, dvSwitch, Attached vNICs
Datastores	Type, Capacity, Free Space, Uncommitted, URL, NAS Host, NAS Path, NAS Username, Local Path	Datacenter, LUNs, Attached Hosts, Attached VMs
LUNs	Disk Name, Disk Key, Device Name, Type, Vendor, Model, UUID, Operational Status	Host, Datastores

Performance monitoring

The following performance metrics are collected every 5 minutes by default. Additional vSphere metrics can be collected by adding them to the appropriate monitoring template.

Component	Metric		
	Name	Derivation	Value type
Clusters	effectiveCPU	clusterServices / effectivecpu	AVERAGE
	effectiveMem	clusterServices / effectivemem	AVERAGE
Hosts	cpuReservedcapacity	cpu / reservedCapacity	AVERAGE
	cpuUsage	cpu / usage	MAXIMUM
	cpuUsagemhz	cpu / usagemhz	MAXIMUM
	diskUsage	disk / usage	MAXIMUM
	memActive	mem / active	MAXIMUM
	memConsumed	mem / consumed	AVERAGE
	memGranted	mem / granted	MAXIMUM
	memSwapused	mem / swapused	MAXIMUM
	sysUpTime	sys / uptime	LATEST
LUNs	diskRead	disk / read	AVERAGE
	diskReadRequests	disk / numberRead	SUMMATION
	diskWrite	disk / write	AVERAGE
	diskWriteRequests	disk / numberWrite	SUMMATION
Host NICs	nicRx	net / received	AVERAGE

Component	Metric		
	Name	Derivation	Value type
	nicTx	net / transmitted	AVERAGE
Resource Pools	cpuUsagemhz	cpu / usagemhz	AVERAGE
	memActive	mem / active	AVERAGE
	memGranted	mem / granted	AVERAGE
VMs	cpuUsage	cpu / usagemhz	AVERAGE
	cpuUsageAvg	cpu / usage	AVERAGE
	cpuUsageMax	cpu / usage	MAXIMUM
	cpuUsageMin	cpu / usage	MINIMUM
	diskUsage	disk / usage	AVERAGE
	memConsumed	mem / consumed	MINIMUM
	memOverhead	mem / overhead	MINIMUM
	memUsage	mem / usage	MINIMUM

Event management

The following vSphere event classes and their subclasses are collected continually.

- Alarm
- Event
- ExtendedEvent
- EventEx

The information in the preceding event classes is used to populate the following Resource Manager event fields.

- device (set to VMware vSphere Endpoint device in the /vSphere device class)
- component
- summary
- severity
- eventClassKey (for mapping specific event types)
- eventKey (for de-duplication and auto-clear fingerprinting

In addition, the following, customized event field is populated for each event.

- description

All vSphere events are timestamped with the time they occur in vCenter, not the time at which they enter the Resource Manager event management system.

Integration with Impact

This ZenPack includes custom state providers for services running on VMware vSphere. Custom state providers enable specialized options for defining state triggers in Zenoss Service Dynamics Service Impact (Service Impact). The following relationships are automatically included in any Service Impact services that contain one or more of

the explicitly mentioned entities. The relationships use the default policy, in which a node inherits the worst state of the ancestor nodes to which it is related.

Internal Service Impact Relationships

- vCenter access failure impacts all datacenters
- Datacenter failure impacts all related hosts, networks, dvSwitches, dvPortgroups and datastores
- Host failure impacts the related cluster or standalone compute resource, and resident VMs
- Host NIC failure impacts the related host
- Network failure impacts related vNICs
- dvSwitch failure impacts related dvPortgroups
- dvPortgroup failure impacts related vNICs
- Datastore failure impacts attached VMs
- Cluster or standalone compute resource failure impacts related resource pools and vApps
- Resource pool failure impacts child resource pools, related vApps and related VMs
- vApp failure impacts related VMs
- vNIC failure impacts the related VM

External Service Impact Relationships

- vNIC failure impacts guest operating system device's related NIC
- VM failure impacts guest operating system device
- NAS file storage providers impact related datastores
- SAN block storage providers impact related datastores
- Resource pool failure impacts related vCloud provider and organizational VDCs
- VM failure impacts related vCloud VM
- Cisco UCS vNIC failure impacts related host NIC
- Cisco UCS service profile failure impacts related host
- VM failure impacts Cisco Nexus 1000V VSM running as guest
- Host failure impacts related Cisco Nexus 1000V VEM
- Host NIC failure impacts related Cisco Nexus 1000V ethernet uplink
- Cisco Nexus 1000V vEthernet impacts related vNIC

A datacenter failure implies that all related hosts are also failed. In some cases this is not appropriate, and the following, customized policies are used.

Custom Service Impact Policies

- vCenter access failure only causes related datacenters to be ATRISK because they're probably still functioning, but may be unmanageable.
- Host NIC failure only implies a host DOWN if all of the host's NICs have failed. If a subset have failed the host is implicitly ATRISK.
- Host failure only implies a cluster DOWN if all of the cluster's hosts have failed. If a subset have failed the cluster is implicitly ATRISK.
- vNIC failure only implies a VM DOWN if all of the VM's vNICs have failed. If a subset have failed the VM is implicitly ATRISK.
- Datastore failure only implies a VM DOWN if all of the VM's datastores have failed. If a subset have failed the VM is implicitly DEGRADED.

Installed items

This ZenPack adds the following items to Resource Manager.

Configuration Properties

zVSphereEndpointHost

zVSphereEndpointUser

zVSphereEndpointPassword

zVSphereEndpointUseSsl

zVSpherePerfQueryChunkSize: How many performance requests to make at a time. Defaults to 50.

zVSpherePerfWindowSize: How many intervals of data to ask for in each performance request. Defaults to 2.

zVSpherePerfDelayCollectionMinutes: How long to lag performance data collection. Defaults to 0.

Device Classes

/vSphere

Modeler Plugins

vmware.vSphere

Datasource Types

VMware vSphere

Monitoring Templates

vSphereCluster (in /vSphere)

vSphereHostSystem (in /vSphere)

vSphereLUN (in /vSphere)

vSpherePnic (in /vSphere)

vSphereResourcePool (in /vSphere)

vSphereVirtualMachine (in /vSphere)

Operational Reports (in the vSphere report organizer)

Clusters

Datastores

Hosts

VMs

VMware Utilization

Collector Daemons

zenvsphere

Adding a vSphere device

Follow these steps to add a vSphere device through the Resource Manager browser interface.

1. Log in to Resource Manager user interface as a user with ZenManager or Manager privileges.
2. Navigate to the **INFRASTRUCTURE** page.
3. From the the **Add** menu, select **Add VMware vSphere Endpoint**.
4. Complete the form with the information for the vSphere device to add.

Name	A name for this endpoint.
Hostname or IP	The host to monitor. The value must be accessible and resolvable from the Collector host.
Username	A user account name the vSphere client accepts.
Password	The password for the user account the vSphere client accepts.

SSL	(checbox) The default is to use SSL communications between this endpoint and the collector host.
Collector	The hostname of the Resource Manager collector host.

5 Click **ADD**.

Adding multiple vSphere devices

Follow these steps to add multiple vSphere devices with the `zenbatchload` command.

1 Log in to the Resource Manager master host as `zenoss`.
2 Create a text file with content similar to the following example.

```
/Devices/vSphere loader='VMware vSphere', loader_arg_keys=['title',
 'hostname', 'username', 'password', 'ssl', 'collector']
vcenter1 host='FQDN-or-IP-address', username='Username',
 password='Password'
```

Replace *FQDN-or-IP-address* with the fully-qualified domain name or IP address of the vSphere device to add, and replace *Username* and *Password* with valid user account information for a vSphere client. You may add multiple endpoints under the same /Devices/vSphere section.

3 Load the devices into Resource Manager. Replace *Filename* with the name of the file created in the preceding step.
`zenbatchload Filename`

Transitioning to this ZenPack

This ZenPack provides a superset of the functionality provided in earlier VMware monitoring ZenPacks, with better performance and reliability.

Note The new and old ZenPacks co-exist gracefully—no automatic migration is attempted, and all history is preserved.

Zenoss recommends transitioning to this ZenPack as soon as possible in all but the following situations:

- You must continue to monitor VMware systems that do not support the vSphere API.

  At some point, Zenoss will discontinue support for the deprecated ZenPacks, so moving older VMware systems to a version that supports the vSphere API is encouraged.
- You must maintain an unbroken history of monitoring, using exactly the same metrics, gathered in exactly the same way.

  Testing or regulatory requirements may prevent an immediate migration. However, no monitoring history is discarded when this ZenPack is enabled. You may maintain the RRD files created with the deprecated ZenPacks alongside the new ones, and use them to examine the complete monitoring history.

To determine whether the deprecated ZenPacks are installed, log in to the broswer interface, and navigate to ADVANCED > ZenPacks. Look for the following ZenPacks.

- ZenPacks.zenoss.EsxTop
- ZenPacks.zenoss.VMwareESXMonitor
- ZenPacks.zenoss.ZenVMware

This ZenPack and the ZenPacks it replaces each add workload to both the monitored device and Resource Manager. Zenoss recommends using both types in parallel only as long as necessary.

Moving VMware systems to this ZenPack

1. Log in to Resource Manager user interface as a user with ZenManager or Manager privileges.
2. Navigate to the **INFRASTRUCTURE** page.
3. In the left column, expand the VMware device class.
4. In the device list, select a vCenter to remove from the class.
5. From the bottom of the left column, click the Action button, and then select **Delete Device...**.
6. In the **Delete Device** dialog, click **SUBMIT**.

Add the deleted vCenter to the `/vSphere` device class. For more information, see *Adding a vSphere device* on page 181.

(WebLogicMonitor) WebLogic Monitor 54

The ZenPacks.zenoss.WebLogicMonitor ZenPack monitors Oracle WebLogic Server services.

This ZenPack uses the JMX Remote API and and accesses MBeans deployed within WebLogic that contain performance information about the components that are being managed. This performance information includes pool sizes for data sources (JDBC), threads, connections (JCA), queues (JMS), servlets, JSPs, Enterprise Java Beans (EJB), timer queues.

Throughput is also monitored when it is available. This metric is computed by WebLogic and is based on the number of messages moving through a queue at any given time. The throughput metric gives a good picture of the health of the messaging subsystem, which is commonly used throughout many enterprise applications. Stateless, Stateful, and Entity EJB performance metrics are monitored, as are message driven bean performance.

Security realms are also monitored for potential denial of service attacks. This includes recording of authentication failures, broken out by valid accounts, invalid accounts, and accounts that are currently locked out. Application specific realms can be monitored by customizing the built in WebLogic default realm.

Overall Application Server Vitals

- Number of total and active JMS connections and servers
- Overall number of JTA transactions that are rolled back or abandoned
- JTA transactions rolled back due to system, application, or resource issues
- Number of JTA rollbacks that timeout
- Active and committed JTA transaction count
- Timer exceptions, executions, and scheduled triggers
- User accounts that are locked and unlocked
- Authentication failures against locked accounts and non-existent accounts
- Total sockets opened, and the current number of open sockets
- JVM Mark/Sweep and Copy garbage collector execution counts
- Number of JVM daemon threads
- JVM Heap/Non-Heap used and committed memory

Entity EJB, Message Driven Bean (MDB), and Session EJB Subsystem Metrics

- Rollback and commit count on a per-EJB basis
- Bean pool accesses, cache hits, and cache misses

- Number of Beans in use, idle, and destroyed
- Number of activations and passivations

Data Pool (JDBC) metrics

- Leaked, Total, and Active connections
- Number of requests waiting for a connection
- Number of reconnect failures

Queue (JMS) Metrics

- Bytes received, currently active, and pending in the queue
- Number of queue consumers
- Number of current, pending, and receives messages

Prerequisites

WebLogic 9.0 or higher is required.

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.ZenJMX, ZenPacks.zenoss.WebLogicMonitor

Configuring WebLogic to Allow JMX Queries

If you have not set up a domain and server then run the `startWLS.sh` script located in the `${BEA_HOME}/wlserver_10.0/server/bin` directory. If you don't have the Terminal I/O package installed you can set the JAVA_OPTIONS variable to the following value:

```
JAVA_OPTIONS="-Dweblogic.management.allowPasswordEcho=true"
export JAVA_OPTIONS
```

Provide a user name and password to start WebLogic. Note that WebLogic requires a password that is at least eight characters long. Wait for WebLogic to generate a configuration and start up. Shut down WebLogic and restart it with remote JMX access enabled.

To enable remote JMX access set the following variable:

```
JAVA_OPTIONS="-Dcom.sun.management.jmxremote.port=12347"
JAVA_OPTIONS="${JAVA_OPTIONS} -
Dcom.sun.management.jmxremote.authenticate=false"
JAVA_OPTIONS="${JAVA_OPTIONS} -Dcom.sun.management.jmxremote.ssl=false"
export JAVA_OPTIONS
```

Then re-run the `./startWLS.sh` script. `JConsole` can then communicate with the server on port 12347.

Configuring Resource Manager

All WebLogic services must have a device entry under the `/Devices/Server/WebLogic` device class.

Note The `zenjmx` daemon must be configured and running. See for more information about configuring the `zenjmx` daemon with the Sun JRE tools.

1 Navigate to the device class or device.

- If applying changes to a device class:
  1 Select the class in the devices hierarchy.
  2 Click **Details**.
  3 Select Configuration Properties.
- If applying changes to a device:
  1 Click the device in the device list.
  2 Select Configuration Properties.

2 Edit the appropriate configuration properties for the device or devices.

Table 46: WebLogic Configuration Properties

Name	Description
zWebLogicJmxManagementAuthenticate	This configuration property is deprecated
zWebLogicJmxManagementPassword	JMX password
zWebLogicJmxManagementPort	The port number used to gather JMX information
zWebLogicJmxManagementUsername	JMX username for authentication

3 Click **Save** to save your changes.

You will now be able to start collecting the WebLogic server metrics from this device.

4 Navigate to Graphs and you should see some placeholders for performance graphs. After approximately 15 minutes you should see the graphs start to become populated with information.

Note The out-of-the-box WebLogic data source configuration has been defined at the macro level, but can be configured to operate on a more granular basis. For example, the Servlet Reload Count applies to all servlets in all web applications but it could be narrowed to be Servlet /submitOrder in web application "production server".

Change the Amount of Data Collected and Graphed

1 Navigate to the device or device class.
2 Select Monitoring Templates in the left panel.
3 From the Action menu, select Bind Templates to display the **Bind Templates** dialog.
4 Move templates from the Available area to the Selected area, and then click **Save**.

Table 47: WebLogic Templates

Name	Description
WebLogic Core	Core information about any WebLogic server, including memory usage, threads, and uptime.
WebLogic JCA	
WebLogic JMS	
WebLogic JMS Destination	
WebLogic JTA	
WebLogic JTA Rollbacks	
WebLogic JVM	
WebLogic Thread Pool	Threadpool metrics measured per Tomcat connector
WebLogic Timer Service	
WebLogic User Lockouts	

5 Click the **OK** button to save your changes.

Viewing Raw Data

For more information about investigating raw data returned from Oracle WebLogic Server, see *Using JConsole to Query a JMX Agent* on page 220.

Monitor SSL-Proxied WebLogic Servers

If you are monitoring a Web application running on WebLogic server, you may find that the transaction always fails with a code 550 regardless of how you configure the script. This could be a result of the WebLogic server being behind an SSL proxy. When used in this configuration, WebLogic requires that a `WL-Proxy-SSL` header be added to the request so that it knows to redirect to HTTPS instead of HTTP.

To support this extra header in your Resource Manager Web transaction, you must make the following changes on the script tab of your WebTx data source.

- Remove any content from the **Initial URL** field.
- Add the following to the beginning of the **Script** box.

```
add_extra_header WL-Proxy-SSL true
go
```

Daemons

Type	Name
Performance Collector	zenjmx

(WebScale) WebScale 55

The ZenPacks.zenoss.WebScale ZenPack adds the `zenwebserver` daemon, to deploy and manage multiple Zope instances.

With this ZenPack, Resource Manager replaces `zopectl` with the `nginx` load balancer.

Prerequisites

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

Installation notes

- The installation process replaces `zopectl` in the startup script (or in the `$ZENHOME/etc/daemons.txt` file) with `zenwebserver`. Use `zenwebserver` to manage the application server.
- If you have multiple Zope instances deployed behind a custom load balancer, installing this ZenPack will not install `zenwebserver` as your control script. You must install it manually after determining and executing your migration strategy.

Usage

zenwebserver Command (Option) Target

Command

- **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.
- **configure** - Generates a new `nginx.conf` file, based on properties in `$ZENHOME/etc/zenwebserver.conf`. The properties and their defaults are shown in the following table.

Property	Default Value	Description
httpPort	8080	Specifies the port to accept the HTTP request.
useSSL	False	Specifies whether SSL config should be used. Set to a value of True to enable.
sslPort	443	Specifies the port to use if useSSL is set.
sslCert	`ZENHOME/etc/ssl/zenoss.crt`	Specifies the path to the SSL certificate if useSSL is set.
sslKey	`ZENHOME/etc/ssl/zenoss.key`	Specifies the path to the SSL key if useSSL is set.

The generated configuration should not be edited. Use the properties in `$ZENHOME/etc/zenwebserver.conf` to customize generation of the `nginx.conf` file.

After running `zenwebserver configure`, you must reload (`zenwebserver reload`) or restart (`zenwebserver restart`) for the new configuration to take effect.

Option

- **-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.

Target

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.

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

```
ngnix
```

- **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
```

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`

Configuring the Load Balancer

The load balancer configuration file (`nginx.conf`) is generated from the template in `$ZENHOME/etc/nginx.conf.template`. This template includes a number of variables that can be substituted by providing values in the `zenwebserver.conf` file.

Custom configurations also can be included in the http and server blocks of the nginx configuration. By default, configuration files in `$ZENHOME/etc` are included if they match one of these patterns:

- `nginx-custom-http-*.conf`
- `nginx-custom-server-*.conf`

Values that can be substituted are:

- Number of worker_processes for nginx to use

```
#worker_processes 4
```

- Paths for nginx var directories

```
#proxy_cache_path $ZENHOME/var/nginx/cache
#proxy_temp_path $ZENHOME/var/nginx/tmp/proxy
#client_body_temp_path $ZENHOME/var/nginx/tmp/client_body
```

- Custom includes, which include any configuration files that match the pattern.
  - customHttpInclude allows configurations to be added to the http block in the nginx configuration:

```
#customHttpInclude $ZENHOME/etc/nginx-custom-http-*.conf
```

  - customServerInclude allows configurations to be added to the server block in the nginx configuration:

```
#customServerInclude $ZENHOME/etc/nginx-custom-server-*.conf
```

- Default error log level

```
#error_log_level warn
```

(WebsphereMonitor) IBM WebSphere 56

The ZenPacks.zenoss.WebsphereMonitor ZenPack monitors IBM WebSphere Application Servers (WAS).

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.ZenWebTx 2.5 or higher, ZenPacks.zenoss.WebsphereMonitor

Configure WAS for Monitoring

To successfully monitor WebSphere, you must have the Performance Monitoring Infrastructure (PMI) servlet installed and enabled on your WebSphere instance. For more information, please see the *IBM WebSphere documentation*.

Configure Resource Manager

1. Navigate to the device or device class under the `/Devices/Server/Tomcat` device class in the Resource Manager interface.
   - If applying changes to a device class:
     1. Select the class in the devices hierarchy.
     2. Click **Details**.
     3. Select Configuration Properties.
   - If applying changes to a device:
     1. Click the device in the device list.
     2. Select Configuration Properties.
2. Edit the appropriate configuration properties for the device or devices.

Table 48: WebSphere Configuration Properties

Property	Description
zWebsphereURLPath	Path to the PMI servlet on a WebSphere instance. The default value is the default path on a WebSphere installation: `wasPerTool/servlet/perfservlet`
zWebsphereUser	Used for HTTP basic authentication. This field is not required, and is empty by default.
zWebspherePassword	Used for HTTP basic authentication. This field is not required, and is empty by default.
zWebsphereAuthRealm	Used for HTTP basic authentication. This field is not required, and is empty by default.
zWebsphereServer	Used by the provided template to build the xpath queries for the data to collect. You must supply a value for this field. There is no default value.
zWebsphereNode	Used by the provided template to build the queries for the data to collect. You must supply a value for this field.

3 Click **Save** to save your changes.
4 Select Device under Monitoring Templates in the left panel.
5 From the Action menu, select Bind Templates.

The Bind Templates dialog appears.

6 Move the Websphere template from the Available list to the Selected list, and then click **Save**.

The **Websphere** template should now be displayed under the **Monitoring Templates for** *Device*. You will now be able to start collecting the WebSphere metrics from this device.

7 Navigate to Graphs and you should see some place holders for graphs. After approximately 15 minutes you should see the graphs start to become populated with information.

Examples

Once the PMI module has been installed into WAS, you can generate the PMI XML file. You then can use this file to complete the monitoring template.

This example shows how to obtain the configuration properties required for basic monitoring functionality. It further shows how to add other metrics to be monitored.

You can generate the PMI XML file by browsing to this URL:

http://*WASserver*/wasPerfTool/servlet/perfservlet

Note This is the default WAS server location. The URL should match the configuration property setting used in the template.

where *WASserver* is the WAS server's host name or IP address.

The following example XML file results:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE PerformanceMonitor SYSTEM "/wasPerfTool/dtd/
performancemonitor.dtd">
<PerformanceMonitor responseStatus="success" version="6.1.0.21">
  <Node name="serverA">
    <Server name="serverAB">
      <Stat name="serverABC">
...
        <Stat name="Dynamic Caching">
          <Stat name="Object: ws/WSSecureMap">
            <Stat name="Object Cache">
              <Stat name="Counters">
  <CountStatistic ID="21" count="0" lastSampleTime="1242827146039"
 name="HitsInMemoryCount" \
    startTime="1242827146039" unit="N/A"/>
  <CountStatistic ID="28" count="5" lastSampleTime="1243610826245"
 name="MissCount" \
    startTime="1242827146039" unit="N/A"/>
              </Stat>
            </Stat>
          </Stat>
        </Stat>
...
       </Stat>
    </Server>
  </Node>
</PerformanceMonitor>
```

In the previous example, configuration properties settings are:

- zWebsphereNode: serverA
- zWebsphereServer: serverAB

You might want to add counters beyond the standard counters. For example, you might want to add the HitsInMemoryCount and MissCount counters (related to dynamic caching). To do this, you would add the following Twill commands to the Script tab of your WebSphere data source:

```
xpathextract HitsInMemoryCount '/PerformanceMonitor/Node[@name="${here/
zWebsphereNode}"]/\
Server[@name="${here/zWebsphereServer}"]/Stat[@name="server"]/
Stat[@name="Dynamic Caching"]/\
Stat[@name="Object: ws/WSSecureMap"]/Stat[@name="Object Cache"]/
Stat[@name="Counters"]/\
CountStatistic[@name="HitsInMemoryCount"]/attribute::count' xpathextract
 MissCount \
'/PerformanceMonitor/Node[@name="${here/zWebsphereNode}"]/\
Server[@name="${here/zWebsphereServer}"]/Stat[@name="server"]/
Stat[@name="Dynamic Caching"]/\
Stat[@name="Object: ws/WSSecureMap"]/Stat[@name="Object Cache"]/
Stat[@name="Counters"]/\
CountStatistic[@name="MissCount"]/attribute::count'
```

After adding these commands, you would then add the data points for HitsInMemoryCount and MissCount, and then add the data points to a graph.

Daemons

Type	Name
Performance Collector	zenwebtx

WindowsMonitor (Microsoft Windows) 57

The ZenPacks.zenoss.WindowsMonitor ZenPack uses WMI to monitor the performance of Microsoft Windows servers.

Note This ZenPack is deprecated; see *(Microsoft.Windows) Microsoft Windows* on page 123.

The `WinPerf` data source uses a Windows performance counter (rather than an SNMP OID) to specify the value to collect. For more information on Windows Management Instrumentation (WMI), please see this *Microsoft Technet Article*.

Table 49: Windows Monitoring Daemons

Name	Description
`zenwin`	Watches Windows services and reports on status.
`zeneventlog`	Watches the Windows event log and generates events.
`zenwinperf`	Collects performance data.

Prerequisites

Supported OS versions are:

- Windows XP
- Windows 2000
- Windows 2003
- Windows Vista
- Windows 2008

Table 50: Windows Performance Monitoring Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x
Required ZenPacks	ZenPacks.zenoss.WindowsMonitor,

Prerequisite	Restriction
	ZenPacks.zenoss.EnterpriseCollector

Defining Windows Credentials

A connection to a Windows device cannot be established without a valid set of credentials. The *zWinUser* and *zWinPassword* configuration properties can be set for each device or for an entire device class.

Note The user needs to be a member of the local administrators or of the domain administrators group unless the steps in are followed.

To set these configuration properties:

1 Navigate to the device or device class in the Resource Manager interface.

- If applying changes to a device class:
  1 Select the class in the devices hierarchy.
  2 Click **Details**.
  3 Select Configuration Properties.
- If applying changes to a device:
  1 Click the device in the device list.
  2 Select Configuration Properties.

2 Edit appropriate configuration properties for the device or devices.

Table 51: Windows Performance Configuration Properties

Name	Description
zWinUser	Windows user with privileges to gather performance information. Like all Windows credentials, the domain should be specified in the *zWinUser* entry. Use `.\username` for an account that is not in the domain but only on the local computer.
zWinPassword	Password for the above user.

3 Click **Save** to save your changes.

Add Devices in Resource Manager

The WindowsMonitor ZenPack includes a /Device/Server/Windows/WMI class that has several device templates bound. SNMP data collection is not used in this class.

To move a device to the /Device/Server/Windows/WMI class:

1 Select the device row in the devices list.
2 Drag the device to the class in the devices hierarchy.

Monitor Other Performance Counters

To create your own `WinPerf` data sources, follow these steps:

1. Navigate to a new or existing monitoring template, and select **New DataSource** from the **Data Sources** table menu.
2. Enter a name for the data source, select **WinPerf** as the type and then click **OK**.
3. Enter a Windows performance counter in the **Perf Counter** field. See *Windows Perfmon counters* for more details.
4. Click **Save**. Notice that a data point is created with the same name as the performance counter you selected.
5. Optionally, test the counter by entering a device ID in the **Test Device** field and clicking the Test button.

Testing Connections from Windows

This procedure verifies that the username/password combination is correct, and that there is no firewall blocking the connection.

1. Run the `wbemtest` command.
2. Click the **Connect…** button.
3. In the **Namespace** field, enter:

```
\\HOST\root\cimv2
```

4. Enter login information in the **User** and **Password** fields.
5. Click the **Query** field.
6. Enter the following query to return a dialog with a list of services on the device.

```
select * from win32_service
```

Testing Connections from Resource Manager

This procedure verifies that the username/password combination is correct, and that there is no firewall blocking the connection. Since this is done from the Resource Manager server, this test is a better approximation of how successful Resource Manager will be in connecting to the Windows device.

As the zenoss user on the Resource Manager server:

```
wmic -U 'user' //device 'select * from Win32_computerSystem'
```

The `wmic` command will then prompt you for the password.

Modify Registry Settings for Firewalls in Secure Environments

Note This procedure is applicable only for environments with firewalls.

The Distributed Component Object Model (DCOM) dynamically allocates one port to each process. You need to decide how many ports you want to allocate to DCOM processes, which is equivalent to the number of simultaneous DCOM processes through the firewall. You must open all of the UDP and TCP ports corresponding to the port numbers you choose. You also need to open TCP/UDP 135, which is used for **RPC End Point Mapping**, among other things. In addition, you must edit the registry to tell DCOM which ports you reserved. You do this with the `HKEY_LOCAL_MACHINES\Software\Microsoft\Rpc\Internet` registry key, which you will probably have to create.

To allow remote registry access for the performance data to be read, see *Controlling remote Performance Monitor access to Windows NT servers.*

The following table shows the registry settings to restrict DCOMs port range to 10 ports.

Table 52: Firewall and Registry Settings for DCOM

Registry Key	Type	Setting
Ports	REG_MULTI_SZ	Range of port. Can be multiple lines such as: 3001-3010 135
PortsInternetAvailable	REG_SZ	Y
UseInternetPorts	REG_SZ	Y

These registry settings must be established in addition to all firewall settings.

Configuring a Standalone Windows Device for a Non-Administrative Account

Monitoring Windows devices normally requires an account with administrator-level privileges. For the Resource Manager user who wants to use a non-administrative account, several additional configuration steps must be performed on each Windows device, or by using a Group Policy.

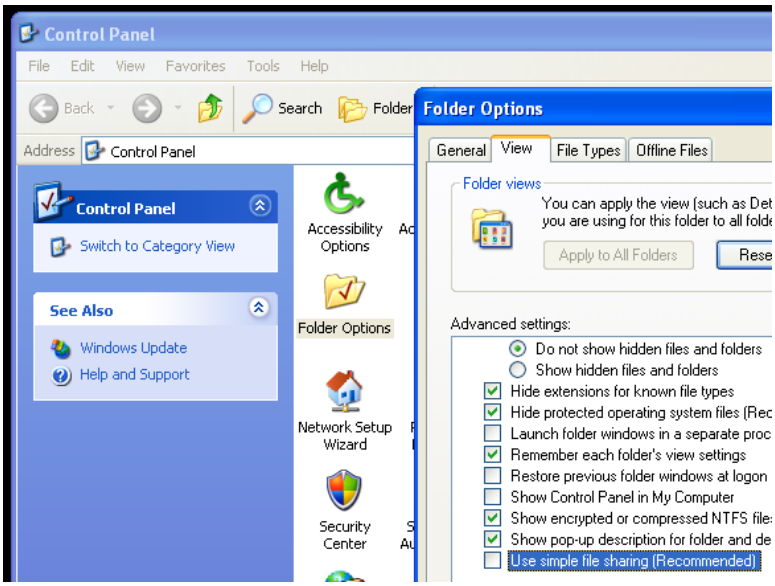
Resource Manager uses the Windows Management Instrumentation (WMI) feature to collect modeling information. The remote Windows registry API also is used to collect low-level performance monitor ("PerfMon") statistics. Both of these Windows sub-systems use the Microsoft Remote Procedure Call (MS-RPC) interface to connect to the Windows device and gather the appropriate information. MS-RPC handles the authentication on a per-packet or per-session basis, but ultimately the access granted is determined by the sub-systems involved with serving the remote procedure calls.

1 If the Windows firewall is in use, modify it to allow Remote Administration access. This will open the MS-RPC port and others as needed. Enter the following command at the command prompt:

```
netsh firewall set service RemoteAdmin enable
```

2 On Windows XP, Simple File Sharing must be disabled for machines that are not located within a Domain. When this feature is enabled it causes all incoming MS-RPC connections to use the built-in Guest account, rather than the account credentials specified in the incoming call. This option may be found by going to **Control Panel**, opening the Folder Options applet and then choosing the **View** tab. In the **Advanced Settings** list, locate the **Use simple file sharing (Recommended)** option, and then disable it.

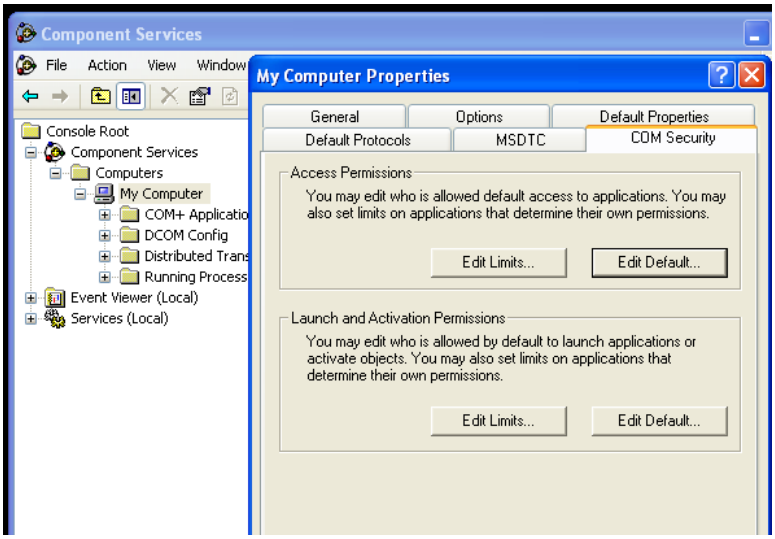
Figure 23: Windows XP Disable Simple File Sharing



3 Create a local account on the Windows device for monitoring. We assume in the remainder of these steps that this account was named `zenossmon` but any valid account name can be used. Place the account only in the Users group and not in the Power Users or Administrators groups. Optionally, create a new user group for monitoring and use that group instead of the account in the remaining steps.

4 Give the `zenossmon` account DCOM access by running the *`dcomcnfg` utility*.

Figure 24: Component Services COM Security Settings



a In the **Component Services** dialog box, expand **Component Services**, expand **Computers**, and then right-click **My Computer** and click **Properties** .

b In the **My Computer Properties** dialog box, click the **COM Security** tab.

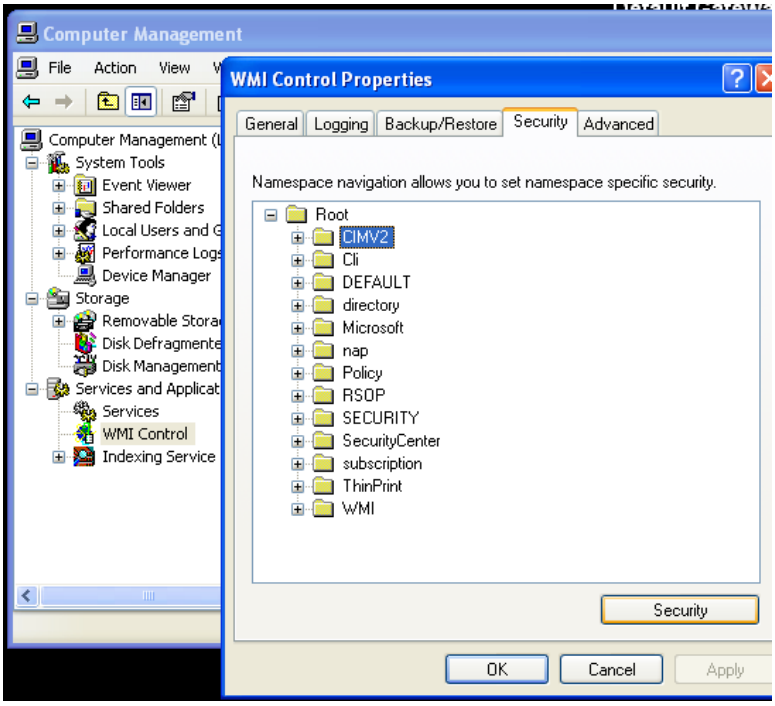
c Under **Access Permissions**, click **Edit Limits**. In the **Access Permission** dialog box, add the `zenossmon` account to the list and ensure that the **Remote Access** checkbox is enabled, then click **OK** to close the dialog.

d Under **Launch and Activation Permissions**, click **Edit Limits**. In the **Access Permission** dialog box, add the `zenossmon` account to the list and ensure that the **Remote Launch** and **Remote Activation** checkboxes are enabled, then click **OK** to close the dialog.

e Click **OK** on the **My Computer Properties** dialog to save all changes.

5 Give the `zenossmon` account permissions to read the WMI namespace by using *WMI Control*.

Figure 25: WMI Control Properties



a Open the **Start** menu and right-click on **My Computer**. Select **Manage** from the menu.

b In the **Computer Management** dialog, expand the **Services and Applications** item and then right-click on **WMI Control**.

c In the **WMI Control Properties** dialog, click the **Security** tab.

d Expand the **Root** namespace, select the **CIMV2** namespace folder and then click **Security**.

e In the **Security for ROOT\CIMV2** dialog, add the `zenossmon` user to the list and ensure the **Enable Account** and **Remote Enable** checkboxes are enabled, then click **OK** to close the dialog.

f In the **WMI Control Properties** dialog click **OK** to close the dialog and save all changes.

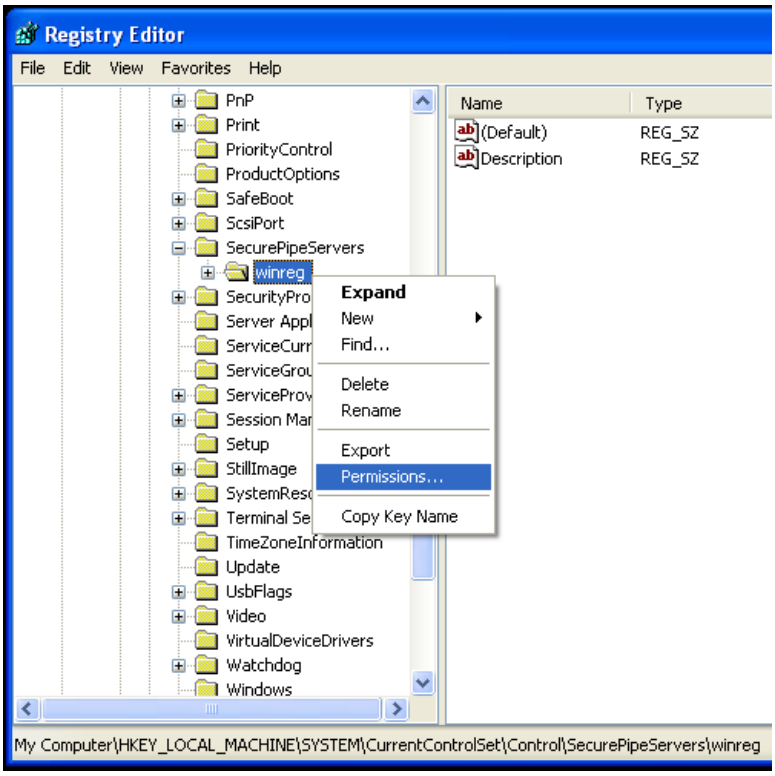
6 At this point in the process remote access to WMI should be enabled and functioning. Test it by running the following command from the Resource Manager server:

```
wmic -U '.\zenossmon' //myhostname 'SELECT Name FROM
 Win32_ComputerSystem'
```

If all is well this command should return the remote system name as the response. If there is any error, carefully recheck the above steps to ensure all access has been properly granted.

7 To gather Windows performance data from PerfMon permissions on the `winreg` registry key must be granted to our monitoring user by using *`regedit`*.

Figure 26: regedit and the winreg Key



a Run `regedit`.

b Browse to the `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurePipeServers\winreg` key.

c Right-click on the `winreg` key and choose **Permissions**.

d Add the monitoring user to the permissions list and grant only `Read` permissions

8 Give the `zenossmon` account *access to read the Windows Event Log*.

Once the appropriate changes are made, test that Event Log access works with your `zenossmon` user. Run the following from your Resource Manager system:

```
wmic -U '.\zenossmon' //myhostname \
'SELECT Message FROM Win32_NTLogEvent WHERE LogFile="Application"'
```

9 If you are using SP1 or newer with Windows Server 2003, then you must *allow non-administrative users to access the service control manager* to monitor services.

At a command prompt, run the following:

```
sc sdset SCMANAGER
D:(A;;CCLCRPRC;;;AU)(A;;CCLCRPWPRC;;;SY)(A;;KA;;;BA)S:(AU;FA;KA;;;WD)
(AU;OIIOFA;GA;;;WD)
```

Note The above command should be one line.

At this point you should be able to query Windows service status remotely by using the non-administrative account. Test this by running the following command from your Resource Manager system:

```
wmic -U '.\zenossmon' //myhostname 'SELECT Name FROM Win32_Service'
```

Tuning Collector Daemon Performance

WindowsMonitor creates several configuration properties that control its behavior. Values for the configuration properties are initially set on the /Devices device class. As with any property, these values can be overridden in other device classes and on individual devices themselves.

Table 53: zenwinperf Daemon Configuration Properties

Property	Setting
zWinPerfCycleSeconds	This is how frequently (in seconds) `zewinperf` data sources are collected. By default this is set to 300 seconds.

Multiple Workers

ZenWinPerf supports multiple workers. This feature allows you to support data collection from more Windows devices without defining additional collectors to host additional ZenWinPerf daemons. The multiple workers feature is enabled by a configuration option:

--workers - Runs ZenWinPerf in a multi-worker setup. By default, set to a value of 2.

Enabling the NTLMv2 Authentication Protocol

To enable the NTLMv2 authentication protocol for all Windows devices of a zenwin, zenwinperf, or zenevent log collector, update collector configuration files:

Alternatively, from the command line add:

```
--ntlmv2auth
```

```
# Enable NTLMv2 authentication for Windows
# Devices, default: False
#ntlmv2auth False
```

58 (XenMonitor) Xen Virtual Hosts

The ZenPacks.zenoss.XenMonitor ZenPack monitors Xen para-virtualized domains.

With this ZenPack, the `zenmodeler` can discover guests running on Xen hosts, and Resource Manager includes screens and templates for collecting and displaying resources allocated to guests.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x
Required ZenPacks	ZenPacks.zenoss.XenMonitor ZenPacks.zenoss.ZenossVirtualHostMonitor

Model Hosts and Guest

For each Xen server, follow this procedure:

1. Optionally, place an SSH key to your Xen server to allow the zenoss user from the Resource Manager server to log in as root without requiring further credentials.
2. Create the Xen server in the /Servers/Virtual Hosts/Xen device class.

   Note If you have this server modeled already, remove the server and recreate it under the Xen device class. Do not move it.

3. Select the **Guest** menu and ensure that the guest hosts were found during the modeling process.

Daemons

Type	Name
Modeler	zenmodeler
Performance Collector	zencommand

Monitoring with sudo

To configure `sudo` in order to run the `xm` on the Virtual Machine Host, you will need to modify a few things.

- Modify the *zCommandPath* zProperty to be blank, otherwise this path will be pre-pended to the `sudo` command.
- Modify the *zCommandUsername* and *zCommandPassword* configuration properties to be a non-root user with `sudo` access to the `xm` command.
- Modify the `Xen.py` modeler to add the `sudo` command. The modeler can be found under the `$ZENHOME/ZenPacks/ZenPacks.zenoss.ZenossVirtualHostMonitor` directory, under the `modeler/plugins` directory.
- Modify the performance templates.

1. Navigate to the `/Devices/Server/Virtual Machine Host/Xen` device class
2. From the device class click on the **Templates** tab
3. Click on the **VirtualMachine** performance template
4. In the **Data Sources** table, click on the **resources** Data Source
5. Edit the command template to add the `sudo` command to the beginning of the `xm` command

59 (ZenDeviceACL) Device ACLs

The ZenPacks.zenoss.ZenDeviceACL ZenPack adds fine-grained device access controls (ACLs) to Resource Manager.

You can use ACLs to limit user access to data, such as limiting access to certain departments within a large organization, or limiting a customer of a service provider to see only his own data.

A user with limited access to objects also has a more limited view of features within the system. Most global views, such as the network map, event console, and all types of class management, are not available. The Device List is available, as are the device organizers Systems, Groups, and Locations. A limited set of reports can also be accessed.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.ZenDeviceACL

Permissions and Roles

Actions in Resource Manager are assigned permissions. For example, to access the device edit screen you must have the "Change Device" permission. Permissions are not assigned directly to a user, but granted to roles, which are then assigned to a user. A common example is the ZenUser role. Its primary permission is "View," which grants read-only access to all objects.

ZenManagers have additional permissions, such as "Change Device," which grants users with this role access to the device edit screen. When you assign a role to a user (using the Roles field on the Edit tab), it is assigned globally. When creating a restricted user you may not want to give that user a global role.

For more information about Resource Manager roles, refer to *Zenoss Service Dynamics Resource Manager Administration*.

Administered Objects

Device ACLs provide limited control to various objects in the system. Administered objects are the same as device organizers (groups, systems, locations, and devices). If access is granted to any device organizer, it extends to all devices in that organizer.

To assign access to objects for a restricted user, you must be assigned the Manager or ZenManager role. Resource Manager grants access to objects by using the “Administered Objects” selection for a user or user group. To limit access, you must not assign a “global” role to the user or group.

Users and Groups

Users and user groups work exactly as they would normally. For more information about managing users and groups, *Zenoss Service Dynamics Resource Manager Administration*.

Assigning Administered Object Access

For each user or group there is selection called "Administered Objects." The Action menu has an "Add" item for each type of administered object. Adding an object will bring up a dialog box with live search on the given type of object.

After adding an object, you can assign it to a role. Roles can be different for each object. For example, a user or group might have the ZenUser role assigned to a particular device but the ZenManager role assigned to a location organizer. If multiple roles are granted to a device though direct assignment and organizer assignment, the resulting permissions will be additive. For the previously cited example, if the device is within the organizer the user will inherit the ZenManager role on the device.

Restricted Screen Functionality

Dashboard

By default, the dashboard is configured with three portlets:

- Object Watch List
- Device Issues
- Production State

These have content that are restricted to objects for a given user.

Device List

The device list is automatically filtered to devices of a restricted user, scoped to accessible devices. There are no menu items available.

Device Organizers

Device organizers control groups of devices for a restricted user. Each device added to the group will be accessible to the user. Permissions are inherited through multiple tiers of a device organizer.

Reporting

Reports are limited to device reports and performance reports.

Viewing Events

A user in restricted mode does not have access to the global event console. The available events for the user can be seen under his organizers.

Create a User Restricted to Specific Devices

1. As admin or any user account with Manager or ZenManager role, create a user named acltest. Set a password for the user.
2. From the user's Edit page, make sure that no role is assigned.
3. Select the user's Administered Objects page.
4. From the Action menu, select the "Add Device…" item and add an existing device to that user.

   The device's role defaults to ZenUser.
5. Log out of your browser, or open a second browser and then log in as acltest.
6. Go to Infrastructure > Devices.

   You should see only the device you assigned to acltest.
7. Navigate to the device and notice that the Edit selection is not available. This is because you are in read-only mode for this device.

Create a Manager Restricted to Specific Devices

Following the previous example:

1. Go back to the acltest user's Administered Objects and set the role on the device to ZenManager.
2. As acltest, navigate to the device. You now have access to the Edit page.

Adding Device Organizers

1. Go to the Groups root and create a group called "RestrictGroup."
2. Go to the acltest user's Administered Objects and add the group to the user.
3. Logged in as acltest, notice that the Navigation menu has the Groups item. Group can be added to a user.
4. Place a device within this group and as acltest you should not only see the device within the group but also in the device list.

Restricted User Organizer Management

1. Assign the acltest user the ZenManager role on your restricted group.
2. As acltest, you can now add sub-organizers under the restricted group.

60 (ZenHoltWinters) Predictive Thresholding

The ZenPacks.zenoss.ZenHoltWinters ZenPack adds the ability to create threshold events when a device exceeds cyclical predicted values.

The *Holt-Winters* exponential smoothing algorithm is used for predictiond. For more information on RRD and Holt-Winters, refer to the `rrdcreate` command.

Note Resource Manager relies on the existence of Holt-Winters RRAs within an RRD file. After adding Holt-Winters thresholds, the RRD files will need to be re-created so that the new configuration can occur. You will have to remove any existing RRD files so that new files can be created.

Removing RRD files will remove all historical information associated with these RRD files.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.ZenHoltWinters

Add a Predictive Threshold

1. Navigate to the template that you want to modify.
2. From the Thresholds area, click (Add Threshold).
3. Provide a name for the new threshold and select the `HoltWintersFailure` threshold type, and then click **Add**.
4. Choose the data source to which the threshold should be applied.
5. Specify the parameters for the prediction engine.

Table 54: Predictive Threshold Data Source Threshold Options

Name	Description
Rows	The number of points to use for predictive purposes.
Alpha	A number from 0 to 1 that controls how quickly the model adapts to unexpected values.

Name	Description
Beta	A number from 0 to 1 that controls how quickly the model adapts to changes in unexpected rates changes.
Season	The number of primary data points in a season. Note that Rows must be at least as large as Season.

6 Click **Save** to save your changes.

7 Remove the RRD file or files that correspond to the data source selected in a previous step.

```
cd $ZENHOME/perf/Devices
rm device_names/DataSource_DataPoint.rrd
```

Note Removing the RRD files does result in a loss of historical information.

61 (ZenJMX) Java Management Extensions

The ZenPacks.zenoss.ZenJMX ZenPack adds the `zenjmx` daemon, which communicates with remote Java Management Extensions (JMX) agents, to collect data from Java-based applications.

This ZenPack defines a data source named `JMX` that allows you to query any single or complex-value attribute, or invoke an MBean operation. It also comes with a built-in template named `Java` that contains MBean information for a few beans built into the JVM.

Note This ZenPack also includes a built-in template named `ZenJMX`. This template should be used only on devices running Java applications that make information available through JMX. To monitor other Java applications, use the included Java template.

When the `zenjmx` daemon is started it communicates with its `zenhub` daemon and retrieves a list of devices that possess `JMX` data sources. It also spawns a Java process. The `zenjmx` daemon asynchronously issues queries for each of those devices to the Java process via XML-RPC. The Java process then collects the data from the Java application to be monitored, and returns the results to the `zenjmx` daemon. Any collection or configuration errors are sent as events to Resource Manager and appear in the event console.

Also, the `zenjmx` daemon sends heartbeat data to its `zenhub` daemon after each collection attempt, to let Resource Manager know it is still alive and well.

JMX Background

The JMX technology is used throughout the Java Virtual Machine to provide performance and management information to clients. Using a combination of `JConsole` (Oracle's JMX client that is shipped with the JDK) and JMX, a system operator can examine the number of threads that are active in the JVM or change the log level. There are numerous other performance metrics that can be gleaned from the JVM, as well as several management interfaces that can be invoked that change the behavior of the JVM.

In Java 5, Oracle introduced the Remote API for Java Management Extensions. This enhancement defines an RMI wrapper around a JMX agent and allows for independent client development. The `zenjmx` daemon accesses remote JMX agents via the Remote API for Java Management Extensions. It currently does not support local connections (provided via the temporary directory) to JMX Agents. JMX also specifies the manner in which various protocols can be used to connect to clients, and send and receive information. The original, most commonly used protocol is RMI. The `zenjmx` daemon supports RMI and JMXMP connections.

ZenJMX Capabilities

The `zenjmx` daemon is a full-featured JMX client that works "out of the box" with JMX agents that have their remote APIs enabled. It supports authenticated and unauthenticated connections, and it can retrieve single-value attributes, complex-value attributes, and the results of invoking an operation. Operations with parameters are also supported so long as the parameters are primitive types (Strings, booleans, numbers), as well as the object version of primitives (such as `java.lang.Integer` and `java.lang.Float`). Multi-value responses from operations (Maps and Lists) are supported, as are primitive responses from operations.

The `JMX` data source installed by this ZenPack allows you to define the connection, authentication, and retrieval information you want to use to retrieve performance information. The IP address is extracted from the parent device, but the port number of the JMX Agent is configurable in each data source. This allows you to operate multiple JMX Agents on a single device and retrieve performance information for each agent separately. This is commonly used on production servers that run multiple applications.

Authentication information is also associated with each JMX data source. This offers the most flexibility for site administrators because they can run some JMX agents in an open, unauthenticated fashion and others in a hardened and authenticated fashion. SSL-wrapped connections are supported by the underlying JMX Remote subsystem built into the JDK, but were not tested in the Zenoss labs. As a result, your success with SSL encrypted access to JMX Agents may vary.

The data source allows you to define the type of performance information you want to achieve: single-value attribute, complex-value attribute, or operation invocation. To specify the type of retrieval, you must specify an attribute name (and one or more data points) or provide operation information.

Any numerical value returned by a JMX agent can be retrieved by Resource Manager and graphed and checked against thresholds. Non-numerical values (Strings and complex types) cannot be retrieved and stored by Resource Manager.

When setting up data points, make sure you understand the semantics of the attribute name and choose the correct Resource Manager data point type. Many JMX Agent implementations use inconsistent nomenclature when describing attributes. In some cases the term "Count" refers to an ever-increasing number (a "Counter" data point type). In other cases the term "Count" refers to a snapshot number (a "Gauge" data point type).

Allowable Parameter Types

The following primitive data types are allowed in `JMX` calls:

- `java.lang.Integer`
- `java.lang.Long`
- `java.lang.Double`
- `java.lang.Float`
- `java.lang.String`
- `java.lang.Boolean`
- `int`
- `long`
- `double`
- `float`
- `boolean`

Single Value Attribute Calls

This is the most basic usage scenario. If you are interested in retrieving a single value from an MBean in a JMX Agent, and the attribute returns simple numeric data, you fall into the "single value attribute" category. To define a

single-value attribute call simply provide the fully qualified name of your MBean and then provide the name of the attribute in the **Attribute Name** field of the data source. Lastly, you must define a data point.

Some examples of this include the commonly referenced JDK Threading information:

- MBean Name: java.lang:type=Threading
- Attribute Name: ThreadCount
- Data Points:
    - ThreadCount (type: gauge)

Java uses lots of file descriptors during normal operation. The number of open file descriptors the JVM is working with can be measured using the following information:

- MBean Name: java.lang:type=OperatingSystem
- Attribute Name: OpenFileDescriptorCount
- Data Points:
    - OpenFileDescriptorCount (type: gauge)

There are several other single-value attributes that can be retrieved from the JDK. We recommend using `JConsole` to interactively navigate through the MBean hierarchy to determine which MBeans contain useful information to you. See for additional information on how to inspect the MBeans deployed in an JMX Agent.

Complex-Value Attribute Calls

If your MBean attribute defines multiple sub-attributes (via CompositeData or Tabular) that you are interested in capturing, then you fall into the category of a "complex-value attribute" call. The JDK contains a few complex-value attributes you might be interested in capturing, including garbage collection statistics that were captured during the copy and mark-sweep compact collection cycles.

To extract data from a complex-value attribute, you must define one or more data points in the data source. The names of the data points are used as keys into the complex-value data structure returned from the MBean attribute. For JMX CompositeData attributes, the data point names are used as a key to map the results. For JMX TabularData, the data point names are used as indexes into the structure to map the result.

The JDK also provides heap memory information via a complex-value attribute. The amount of committed, used, and maximum heap memory can be viewed by setting up a complex-value attribute in Resource Manager with the following information:

- MBean Name: java.lang:type=Memory
- Attribute Name: HeapMemoryUsage
- Data Points:
    - committed (type: gauge)
    - used (type: gauge)
    - max (type: gauge)

Example Method Calls

Some management values need to be computed. These situations frequently arise when custom MBeans are deployed alongside an enterprise application. An MBean named "Accounting" might be deployed within an enterprise application that defines operations intended for operators or support staff. These operations might include methods such as "getBankBalance()" or "countTotalDeposits()".

The `zenjmx` daemon can invoke operations, but there are some subtleties in how it sends parameters to a JMX Agent, and interprets the response.

No parameters, single return value

In the most basic usage scenario no arguments are passed to the operation and a single value is returned. This usage scenario is very similar to a single-value attribute call, except we're invoking an operation to retrieve the value rather than accessing an attribute. The configuration for this hypothetical usage scenario follows:

- MBean Name: Application:Name=Accounting,Type=Accounting
- Operation Name: getBankBalance()
- Data Points:
    - balance (type: gauge)

No parameters, multiple values returned in List format

In this scenario no parameters are passed to an operation, but multiple response values are provided in a List. The values returned are expressed in a List<Object>, but they are coerced (but not casted) to doubles prior to being stored in Resource Manager. This means that returning a numeric value as "1234" will work, but "1,234" will not work. The litmus test is to evaluate if `Double.valueOf(object.toString())` will successfully evaluate.

The `zenjmx` daemon can be configured to read multiple values from an operation's results by defining multiple data points. You must define a data point for each value returned from the operation, and if there is a mismatch between the number of data points you define and the size of the List<Object> returned an exception will be generated. The configuration for the `zenjmx` daemon follows:

- MBean Name: Application:Name=Accounting,Type=Accounting
- Operation Name: getBalanceSummary()
- Data Points:
    - dailyBalance (type: gauge)
    - annualBalance (type: gauge)

No parameters, multiple values returned in Map format

In this scenario no parameters are passed to an operation, but multiple response values are provided in a Map<String, Object>. The keyset of the Map contains the names of data points that can be defined, and the values are the values of said data points. When a Map<String, Object> is returned you need not capture all of the returned values as data points, and you can instead pick the exact values you are interested in. To choose the values to capture you simply define data points with the same names as Strings in the keyset.

The following configuration demonstrates how to extract specific data points from an operation that returns a Map<String, Object>. The key item to note in this configuration is that "dailyBalance" and "annualBalance" must be present as keys in the returned Map<String, Object> and their values must be coercible via the Double.valueOf(object.toString()) idiom.

- MBean Name: Application:Name=Accounting,Type=Accounting
- Operation Name: getBalances()
- Data Points:
    - dailyBalance (type: gauge)
    - annualBalance (type: gauge)

Single parameter in polymorphic operation

MBeans are implemented as Java classes and Java permits parameterized polymorphic behavior. This means that multiple methods can be defined with the same name so long as their parameter signatures differ. You can safely define "getBalance(String)" and "getBalance()" and the two exist as separate methods.

In order to properly resolve methods with the same name the caller must provide a Class[] that lists the types of parameters that exist in the method's signature. This resolves the candidate methods to an individual method which can then be invoked by passing an Object[].

The `zenjmx` daemon allows you to resolve methods of the same name and asks you to provide the fully qualified class names of each parameter in comma delimited format when you set up the data source. Note that primitive types (String, Boolean, Integer, Float) are supported but complex types are not supported, and that you must include the class' package name when providing the information (java.lang.String).

The Object[] of parameter values must line up with Class[] of parameter types, and if there is a mismatch in the number of types and values that are provided an exception will be generated.

The marshaling of values from String to Boolean, Integer, and Float types is provided via the .valueOf() static method on each of those types. That is, if you define an attribute of type java.lang.Integer you must provide a String that can be successfully passed to java.lang.Integer.fromValue(). If you fail to do so an exception is generated.

This example illustrates how to pass a single parameter to a polymorphic operation:

- MBean Name: Application:Name=Accounting,Type=Accounting
- Operation Name: getBalances()
- Paramater Types: java.lang.Integer
- Parameter Values: 1234
- Data Points:
    - balance (type: gauge)

Here is another example where we've changed the type of the parameter passed to the method to be a String. Semantically it represents a different type of Account in our example:

- MBean Name: Application:Name=Accounting,Type=Accounting
- Operation Name: getBalances()
- Paramater Types: java.lang.String
- Parameter Values: sbb552349999
- Data Points:
    - balance (type: gauge)

Multiple parameters in polymorphic operations

The preceding example describes how polymorphic behavior in Java functions and how method resolution can be provided by identifying the Class[] that represents the parameters passed to a method. The situation where multiple parameters are passed to a polymorphic operation is no different then the situation where a single parameter is passed to a polymorphic operation, except that the length of the Class[] and Object[] is greater than one.

When multiple parameters are required to invoke an operation you must provide the fully qualified class names of each parameter's type in comma delimited format, as well as the object values for each type (also in comma delimited format).

The following example demonstrates a configuration that passes two parameters to an MBean operation. The second parameter passed is a default value to return if no account can be located matching the first parameter.

- MBean Name: Application:Name=Accounting,Type=Accounting
- Operation Name: getBalances()
- Parameter Types: java.lang.String, java.lang.Integer
- Parameter Values: sbb552349999, 0
- Data Points:
  - balance (type: gauge)

There are additional combinations that are possible with polymorphic methods and the values they return, and those combinations are left as an exercise for the reader to explore. The logic for extracting results from multi-value operation invocations follows the same rules as the logic for extracting results from a multi-value attribute read. For additional information on the rules of that logic see the section above on multi-value attributes.

Special Service URLs

By default, URLs are assembled as:

```
service:jmx:rmi:///jndi/rmi://hostName:portNum/jmxrmi
```

This host name and port points to a registry. After a JMX agent connects to the registry, the registry tells the agent which host and port to use for remote calls.

In some situations, you may want to explicitly provide the registry host and port, as well as the host and port for the remote calls. Use the long form, as in:

```
service:jmx:rmi://127.0.0.1:8999/jndi/rmi://127.0.0.1:8999/jmxrmi
```

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.ZenJMX
Other	Oracle JRE Version 5.0 or higher is required.

Oracle Java Runtime Environment (JRE)

This ZenPack requires Oracle JRE Version 5.0 or higher. Make sure that after you install the JRE you update your PATH such that the `java` executable works. You can test this using the command:

```
$ which java
/usr/java/default/bin/java
```

If the above returns a fully qualified path, then you have successfully installed Java.

If Java is not installed, the `which` will return a message similar to the following:

```
$ which java
/usr/bin/which: no java in (/usr/local/bin:/bin:/usr/bin:/opt/zenoss/bin)
```

To determine which version of Java is installed, run the following command:

```
$ java -version
java version "1.5.0_16"
Java(TM) 2 Runtime Environment, Standard Edition (build 1.5.0_16-b06-284)
Java HotSpot(TM) Client VM (build 1.5.0_16-133, mixed mode, sharing)
```

Note Oracle Java is required. Other Java implementations do not work.

Example to Monitor a JMX Value

Enabling Remote JMX Access

Each application server has a slightly different process for enabling remote JMX Access. You should consult with your application server for specific instructions. This section includes instructions for a few commonly used configurations.

JMX agents can be configured in two ways: remote access and local-only. When configured for remote access a JMX client communicates with the JMX agent via a socket and uses a remote protocol such as Remote Method Invocation (RMI) or JMXMP to access the MBeans. When configured for local-only access the JMX agent periodically dumps serialized MBeans to a temporary directory on the machine. `JConsole` can be used to access JMX agents in local-only mode as well as in remote mode. The `zenjmx` daemon can be used only with remote servers via RMI or JMXMP and cannot work with local-only serialized MBeans. This is not a significant limitation because the `zenjmx` daemon can establish RMI connections to localhost in the same manner that it creates connections to remote hosts.

The JAVA_OPTS environment variable can be used to enable remote access to JVM MBeans. Set it as follows:

```
JAVA_OPTS="-Dcom.sun.management.jmxremote.port=12345
JAVA_OPTS="${JAVA_OPTS} -
Dcom.sun.management.jmxremote.authenticate=false"
JAVA_OPTS="${JAVA_OPTS} -Dcom.sun.management.jmxremote.ssl=false"

export JAVA_OPTS
```

When starting an application pass the JAVA_OPTS variable as an argument to the JVM as follows:

```
java ${JAVA_OPTS} -classpath /path/to/application.jar
 com.yourcompany.Main
```

You can then use `JConsole` to connect to localhost:12345. Authentication can be configured by modifying the java.security file as well as java.policy. There are lots of examples available on the Internet that can provide guidance in how to achieve authenticated remote access to JVM MBeans.

Configure Resource Manager with a Custom Data Source

Custom JMX data sources allow system administrators to monitor any attribute or operation result accessible via a JMX call. This ZenPack creates a `JMX` data source and allows you to provide object information, as well as authentication settings, and attribute/operation information. Determining which object and attribute names, as well as which operations to invoke, is the key to customizing this feature.

To configure the system with a custom data source:

1 Select Infrastructure from the navigation bar.

2 Click the device in the device list.

The device overview page appears.

3 Expand Monitoring Templates in the left panel, and then select Device.

4 Select Add Local Template from the Action menu.

The Add Local Template dialog appears.

5 Enter a name for the template (such as JVM Values), and then click **Submit**.

The template is added.

6 Select the newly created template.

7 Click Add in the Data Sources area.

The Add Data Source dialog appears.

8 Enter a name for the data source (Heap Memory), select JMX as the type, and then click Submit.

The data source is added.

9 Double-click the data source to edit it. Change options as needed, and then click **Save**.

Table 55: Memory Head Example ZenJMX Data Source Options

Option	Description
Protocol	RMI or JMXMP. Consult your Java application documentation to determine which JMX Connector protocols it supports.
JMX Management Port	This is not necessarily the same as the listen port for your server.
Object Name	The Object Name is also referred to as the MBean name. Enter `java.lang:type=Memory`
Attribute Name	Enter `HeapMemoryUsage`

10 Add data points named `committed`, `max`, and `used`:

a Select Add Data Point from the Action menu.

The Add Data Point dialog appears.

b Enter the name of the data point (`committed`, `max`, or `used`) and then click **Submit**.

11 After adding all data points, add graphs that reference them.

Review to learn how to determine the object name, attribute name, and data points that might be interesting in your application.

Monitor Values in TabularData and CompositeData Objects

The Attribute Path input value on the ZenJMX data source allows you to monitor values nested in the TabularData and CompositeData complex open data objects. Using this value you can specify a path to traverse and index into these complex data structures.

If the result of traversing and extracting a value out of the nested open data is a single numeric value then it is automatically mapped to the datapoint in the data source. However, if the value from the open data is another open data object then the data point names from the datasource are used as indexes or keys to map values out of the open data.

The input value is a dot-separated string that represents a path through the object. Non-bracketed values are keys into CompositeData. Bracketed values are indexes into TabularData.

For TabularData indexes with more than one value, use a comma-separated list with no spaces (for example, [key1,key2]).

To specify a column name (needed only when the table has more than two columns) use curly brackets after the table index.

Example

To get the used Tenured Generation memory after the last garbage collection from the Garbage Collector MBean, set the Attribute Name on the datasource to lastGcInfo. Set the Attribute Path to:

```
memoryUsageAfterGc.[Tenured Gen].{value}.used
```

The key `memoryUsageAfterGc` is evaluated against the CompositeData returned from the `lastGcInfo` attribute. The evaluation results in a TabularData object. Then, the `[Tenured Gen]` index is evaluated against the TableData, which returns a row in the table.

Since a row in the table can contain multiple columns, the key `value` (in curly brackets) is used to pick a column in the row. Lastly, the key `used` is evaluated against the CompositeData in the column to return the memory value.

In this example, since the index being used for the tabular data is not a multi-value index and so the column name is optional. The Attribute Path can be written as:

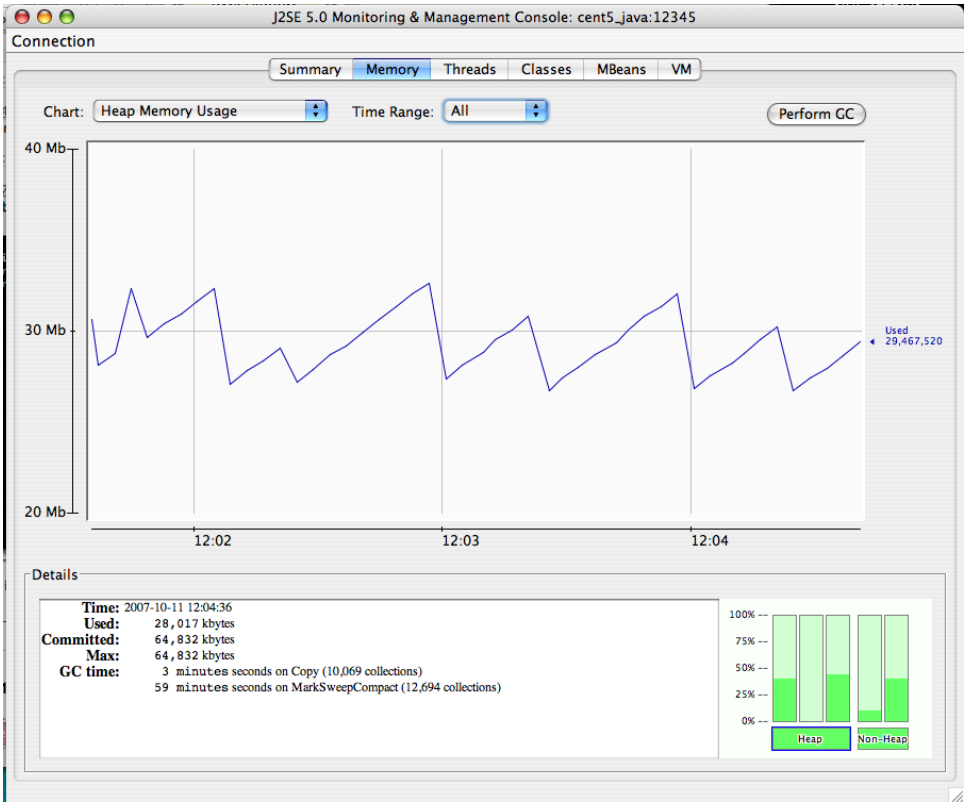
```
memoryUsageAfterGc.[Tenured Gen].used
```

Using JConsole to Query a JMX Agent

`JConsole` is a tool built into the JDK that allows system administrators to query a JMX Agent and examine the MBeans deployed within the server. `JConsole` also allows administrators to view JVM summary information, including the amount of time the JVM has been running, how many threads are active, how much memory is currently used by the heap, how many classes are currently loaded, and how much physical memory exists on the machine.

`JConsole` also provides a graph that shows memory, thread, and class usage over time. The scale of the graph can be adjusted so that a system administrator can examine a specific period of time, or can zoom out to view a longer range picture of usage. Unfortunately, `JConsole` can only produce graphs that show usage while `JConsole` was running. Administrators cannot look back in time to a point where the JVM was running but `JConsole` was not monitoring the JVM.

Figure 27: JMX Heap Graph

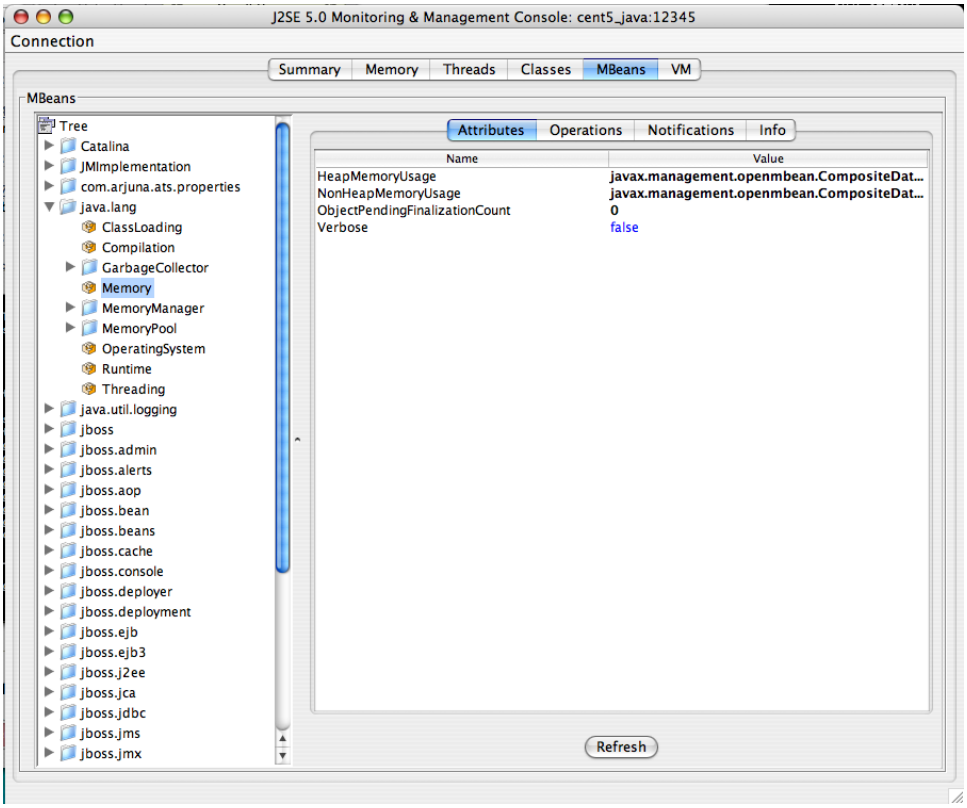


The **MBeans** tab along the top of `JConsole` provides an interactive method for examining MBean values. After clicking on the **MBeans** tab a panel will be displayed with a tree on the left hand side. The tree contains a hierarchical list of all MBeans deployed in the JVM.

The standard JVM MBeans are all in the java.lang and java.util.logging packages. Application server specific MBeans do not follow any standard naming pattern. Some vendors choose to use package names for their MBean names while other vendors choose package-like names (but not fully qualified packages).

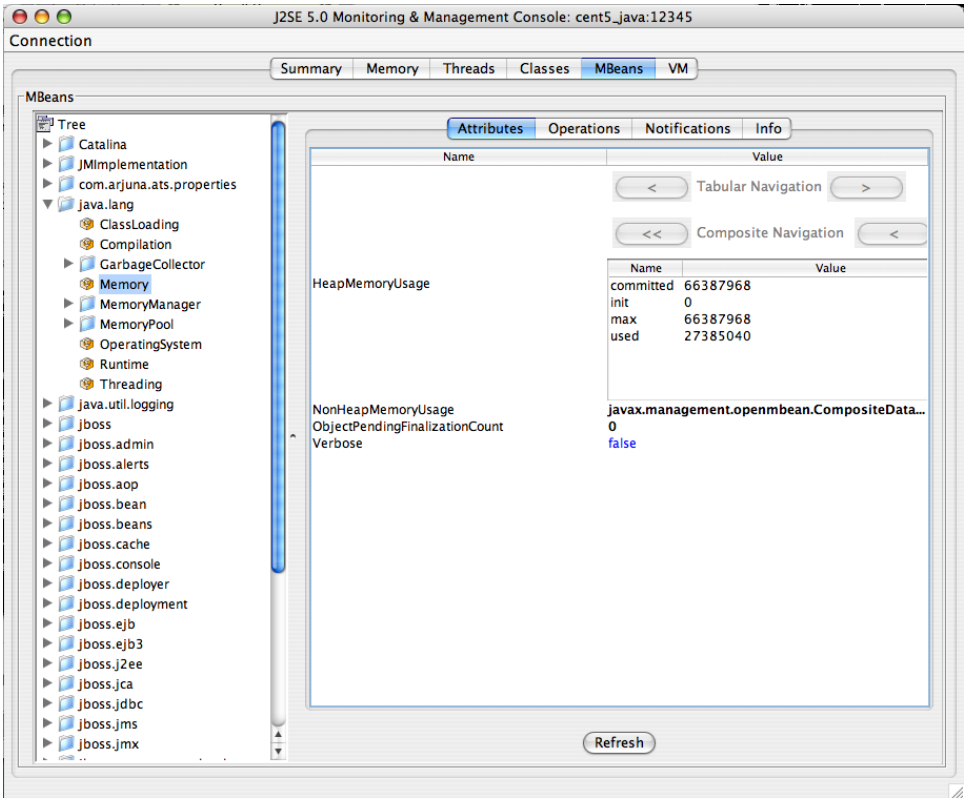
To get started expand the java.lang node in the Tree. This will expose several MBeans as well as additional folders. Click on the Memory MBean and observe how the right hand side of the panel is populated with information about the Memory MBean.

Figure 28: Memory MBean



MBeans can contain attributes and operations. MBeans can also fire notifications to observers, but that's beyond the scope of this document. The attributes tab lists all of the attributes in the first column and their values (or a clickable attribute type) in the second column. In the case of Memory the HeapMemoryUsage is a Composite attribute, otherwise referred to as a "complex-value attribute" in Resource Manager. Double click the "javax.management.openmbean.CompositeDataSupport" type and you will see multiple attributes appear. The show the amount of committed, maximum, and used memory sizes for the heap.

Figure 29: Memory MBean Expanded

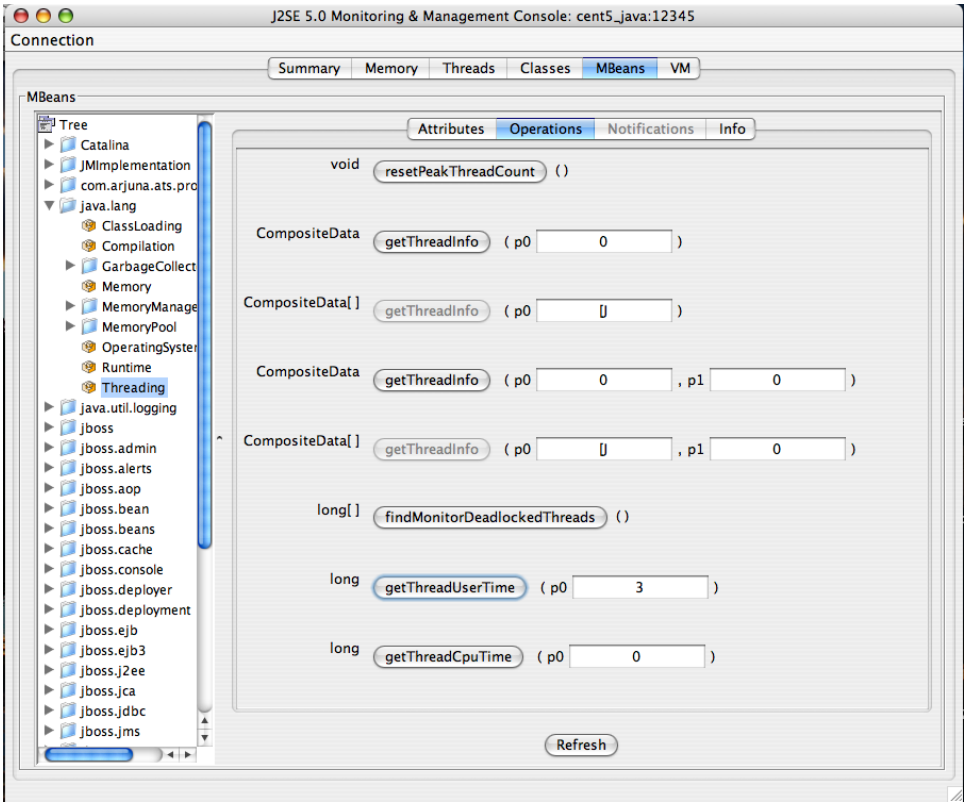


The unique name of the MBean can be viewed by clicking on the Info tab. The first value is MBean Name. Its value in the case of Memory is: "java.lang:type=Memory."

Note There is no standardized way to name MBeans; application server vendors name them differently.

You can also examine operation information by clicking on the Operations tab. These are methods that `JConsole` can remotely invoke on an MBean that will result in some value being computed or some state changing in the application. The Threading MBean has several operations that can be invoked that return information. Click on the java.lang package and then click on the Threading operation. Lastly, click on the Operations tab. Methods like "getThreadUserTime" are invocable.

Figure 30: Operations Tab



Test the "getThreadUserTime" method by changing the p0 parameter to 1 and clicking the "getThreadUserTime" button. A dialog window will be raised that displays the amount of CPU user time thread #1 has used. Try adjusting the parameter to different values to observe the different CPU times for the threads.

zenjmx Options

To display the options supported by the `zenjmx` daemon, enter the following command:

```
zenjmx help
```

Memory Allocation

Use the `--javaheap` option to set the max heap. The value is 512MB.

ZenJMX Logging

You can adjust logging levels to reduce the size of ZenJMX log files. In the `log4j.properties` file (in `$ZENHOME/Products/ZenJMX`), update the first line and change DEBUG to INFO, WARN, or ERROR.

Daemons

Type	Name
Performance Collector	`zenjmx`

62 (ZenMailTx) Mail Transactions

The ZenPacks.zenoss.ZenMailTx ZenPack allows you to monitor round-trip email delivery.

Events

There are several situations in which this ZenPack creates events. The component will be `zenmailtx`, the eventGroup will be `mail` and the eventClass will be `/Status`. These situations are:

- The SMTP server name or the POP server name cannot be resolved.
- The SMTP server or the POP server is down or unavailable.
- The timeout (specified on the Data Source tab) is exceeded for the SMTP or POP server.
- Authentication (if specified) with the SMTP or POP server fails.
- A threshold defined for one of the data points in this data source is exceeded. Thresholds are defined in the monitoring template that contains the data source.

Once an email has successfully made a trip back and forth, a clear event is created that clears any failure events.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.ZenMailTx

Enable Monitoring

1. Click the device in the device list.
2. From the left panel, select the Device template under Monitoring Templates.
3. Select Add Local Template from the Action menu.
4. Enter an identifier for the template (such as ZenMailTx), and then click **Submit** to create the template.
5. Click the newly created ZenMailTx template.
6. In the Data Sources area, click Add to add a data source.
7. Enter a name for the data source (MailTx), select MAILTX as the type, and then click **Submit**.
8. Change options as needed.

Table 56: Mail Transactions Basic Data Source Options

Option	Description
To Address	The recipient e-mail address. This should be the same as the POP Username.
From Address	The e-mail address that will appear in the From: field in the generated e-mail
SMTP Host	The e-mail server used by Resource Manager to send the email
POP Host	The email server where you will retrieve your test message

Note Any of the `MAILTX fields` can take TAL expressions, including the password fields.

9 Click **Save** to save your changes.

10 Navigate to Graphs and you should see some place holders for graphs. After approximately fifteen minutes you should see the graphs begin populating with information.

Daemons

Type	Name
Performance Collector	zenmailtx

Managing the Collector Daemon

The zenmailtx daemon:

- Sends the test email message via the specified SMTP server
- Retrieves the email message from the specified POP server
- Sends the following information to Resource Manager:
  - Time taken to send
  - Time taken to fetch
  - Total time

This daemon appears on the Resource Manager Daemons page and can be started, stopped and restarted from there.

63 (ZenOperatorRole) Operator Role

The ZenPacks.zenoss.ZenOperatorRole ZenPack adds the `ZenOperator` user role.

For more information about using this role, refer to *Zenoss Service Dynamics Resource Manager Administration*.

Prerequisites

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.ZenOperatorRole

64 (ZenSQLTx) SQL Transactions

The ZenPacks.zenoss.ZenSQLTx ZenPack monitors the availability and performance of MySQL, Sybase and Microsoft SQL servers.

With this ZenPack, user-defined SQL queries can be executed against a database, and the results are returned as a SQL data source.

Prerequisites

Each remote collector must have an installed MySQL client.

Prerequisite	Restriction
Product	Resource Manager 4.x, Zenoss 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.ZenSQLTx
MySQL client	Each remote collector must have an installed MySQL client.

Enable SQL Server Monitoring

Ensure that your Microsoft SQL Server authentication mode is set to "SQL Server and Windows Authentication mode." For more information about this setting and how to change it, refer to:

http://msdn.microsoft.com/en-us/library/ms188670.aspx

1. Click the device in the device list.
2. Select Device under Monitoring Templates in the left panel.
3. Select Add Local Template from the Action menu.

   The Add Local Template dialog appears.
4. Enter a name of the template, and then click **Submit**.
5. Click the newly created template in the left panel.
6. In the Data Sources area, click Add.
7. Enter a name for the data source, select `SQL` as the type, and then click **Submit**.
8. Double-click the newly created data source.

   The Edit Data Source dialog appears.

9 Change options as needed.

Table 57: MS SQL Server Transactions Data Source Options

Option	Description
Database Type	Enter `MS SQL`
Host Name	Set the host name on which the database is located. This field accepts a TALES expression, such as `${here/id}` or `${here/getManageIp}`
Port	Set the port on which the database server is listening. If you do not specify a port number, then the default port for the database is used.
Database Name	Specify the name of the database (required).
User	Specify a user name with permission to connect to the database and run queries.
Password	Specify the user password.
SQL Queries	Specify the SQL queries that this data source should execute. A summary of MS SQL syntax is available in the documentation accompanying the software.

10 Click **Save** to save your changes.

Resource Manager creates a data point that corresponds to the total query time in milliseconds.

11 Click **Test** to verify that the database connection can be completed, and that the data returned from the queries are correct.

For more information about setting up thresholds, graphs, and data points, refer to *Zenoss Service Dynamics Resource Manager Administration*.

Enable Sybase Server Monitoring

1 Click the device in the device list.
2 Select Device under Monitoring Templates in the left panel.
3 Select Add Local Template from the Action menu.

The Add Local Template dialog appears.

4 Enter a name of the template, and then click **Submit**.
5 Click the newly created template in the left panel.
6 In the Data Sources area, click Add.
7 Enter a name for the data source, select `SQL` as the type, and then click **Submit**.
8 Double-click the newly created data source.

The Edit Data Source dialog appears.

9 Change options as needed.

Table 58: MySQL Server Transactions Data Source Options

Option	Description
Database Type	Enter `Sybase`

Option	Description
Host Name	Set the host name on which the database is located. This field accepts a TALES expression, such as `${here/id}` or `${here/getManageIp}`
Port	Set the port on which the database server is listening. If you do not specify a port number, then the default port for the database is used.
Database Name	Specify the name of the database (required).
User	Specify a user name with permission to connect to the database and run queries.
Password	Specify the user password.
SQL Queries	Specify the SQL queries that this data source should execute. A summary of Sybase syntax is available at the *Sybase Manuals* Web site.

10 Click on the **Save** button to save your changes.

Resource Manager creates a data point that corresponds to the total query time in milliseconds.

11 Click **Test** to verify that the database connection can be completed, and that the data returned from the queries are correct.

For more information about setting up thresholds, graphs, and data points, refer to *Zenoss Service Dynamics Resource Manager Administration*.

Enable MySQL Server Monitoring

1 Click the device in the device list.
2 Select Device under Monitoring Templates in the left panel.
3 Select Add Local Template from the Action menu.

The Add Local Template dialog appears.

4 Enter a name of the template, and then click **Submit**.
5 Click the newly created template in the left panel.
6 In the Data Sources area, click Add.
7 Enter a name for the data source, select `SQL` as the type, and then click **Submit**.
8 Double-click the newly created data source.

The Edit Data Source dialog appears.

9 Change options as needed.

Table 59: MySQL Server Transactions Data Source Options

Option	Description
Database Type	Enter `MySQL`
Host Name	Set the host name on which the database is located. This field accepts a TALES expression, such as `${here/id}` or `${here/getManageIp}`
Port	Set the port on which the database server is listening. If you do not specify a port number, then the default port for the database is used.

Option	Description
Database Name	Specify the name of the database (required).
User	Specify a user name with permission to connect to the database and run queries.
Password	Specify the user password.
SQL Queries	Specify the SQL queries that this data source should execute. A summary of MySQL syntax is available at: *http://dev.mysql.com/doc/refman/5.0/en/sql-syntax.html*

10 Click on the **Save** button to save your changes.

Resource Manager creates a data point that corresponds to the total query time in milliseconds.

11 Click **Test** to verify that the database connection can be completed, and that the data returned from the queries are correct.

For more information about setting up thresholds, graphs, and data points, refer to *Zenoss Service Dynamics Resource Manager Administration*.

Storing Query Results

If any data is retrieved from the database that can be interpreted as a number, that number can be used as a data point. In select statements in which a column name is used, that column name becomes the name of the data point. In select statements in which no column name is specified (for example, aggregate functions such as `count(*)`, `sum()`, or `min()`), the data point name returned is database-dependent:

- MySQL - The column name can be controlled with an 'AS' clause in the query. If used, then the column name is the "cleaned up" result of the 'AS' clause; otherwise, it uses the format: 'q' + query number (beginning with 0) + '_' + column number in the query (beginning with 0).
- All other databases - The column name uses the format: 'q' + query number (beginning with 0) + '_' + column number in the query (beginning with 0).

Non-alphanumeric characters ([^za-zA-Z0-9_]) are removed from the column name to produce the data point name. Any query results that cannot be interpreted as a number are ignored, and the query numbers will not change.

For example, the queries:

```
select count(*) from Users;select UserName from Users; select count(*) *
 4 from Users
```

return these results:

```
Queries completed successfully. | totalTime=2.13289260864 count=3.0
 count4=12.0
```

Note To use multiple queries (such as in the preceding example), they must be separated with a semicolon.

This example demonstrates multiple results from a single query:

```
select count(*) as count1, count(*)-1001 from history;
```

and returns these results:

```
Queries completed successfully. | totalTime=72.6099014282 count1=99894.0
 count1001=98893.0
```

Notes:

- For SQL Server, use the format `q*_*` if no column name is found.
- The SQL 'as' renaming capability can be used to control the name of the data point.

Troubleshooting

To verify any queries, as well as any permissions or authentication issues, run the `zensql.py` command from the command line. Here's an example against the MySQL database on a Resource Manager server:

```
cd $ZENHOME/ZenPacks/*ZenSQLTx*/Z*/z*/Z*
./zensql.py -t mysql -H localhost -u zenoss -p zenoss -d events 'select
 \* from events.log;'
Queries completed successfully. | totalTime=54.5899868011
```

Note Single quotes (') are required around the SQL statement. Any wild card characters (such as `*`) must be escaped, as shown in the previous example.

For the `zensql.py` command, the database types understood are shown in the following table.

Table 60: zensql.py Database Types

Name	Database Type
`mssql`	MS SQL Server
`sybase`	Sybase
`mysql`	MySQL Server

Daemons

Type	Name
Performance Collector	zencommand

(ZenVMware) VMware vSphere 65

The ZenPacks.zenoss.ZenVMware ZenPack monitors VMware devices through the vSphere API.

Note This ZenPack is deprecated; see *(vSphere) VMware vSphere* on page 177.

By entering a single set of connection parameters, Resource Manager can :

- Obtain the names and properties of various entities in your VMware infrastructure
- Monitor metrics collected by VMware
- Retrieve VMware events

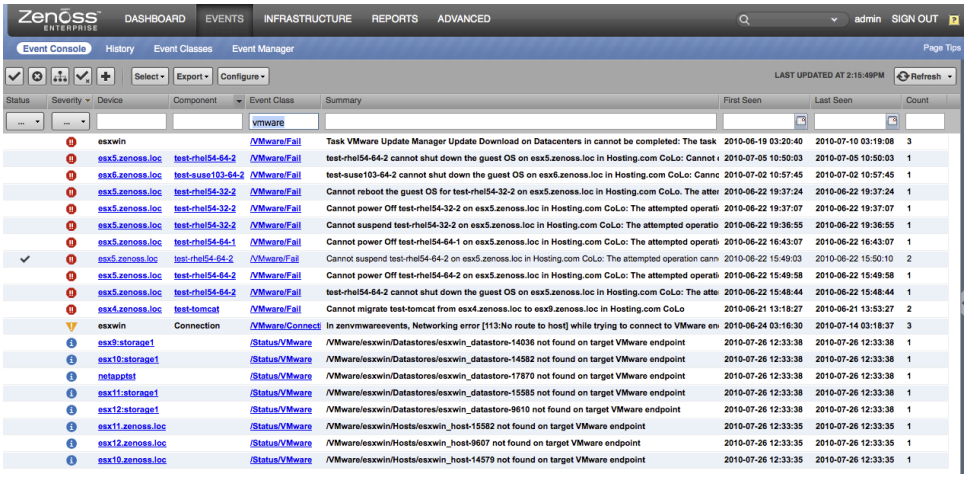
Resource Manager extracts VMware information through the VMware Infrastructure (VI) SDK, VMware's SOAP interface to its line of server virtualization products. The SDK can be accessed from an individual ESX server or vCenter Server (previously, VirtualCenter Server) instance, which can return information about many ESX servers.

For more information about VMware infrastructure, see VMware's *Virtualization* overview.

VMware Events

VMware records a wide range of events that are available through the VI SDK. Resource Manager extracts these events and makes them available in the event console.

Figure 31: VMware Events (Event Console)



The device column shows the ID of the VMware entity with which the event is associated, unless the event is specific to a guest VM. In that case, the Device column shows the ID of the host, and the Component column displays the ID of the guest.

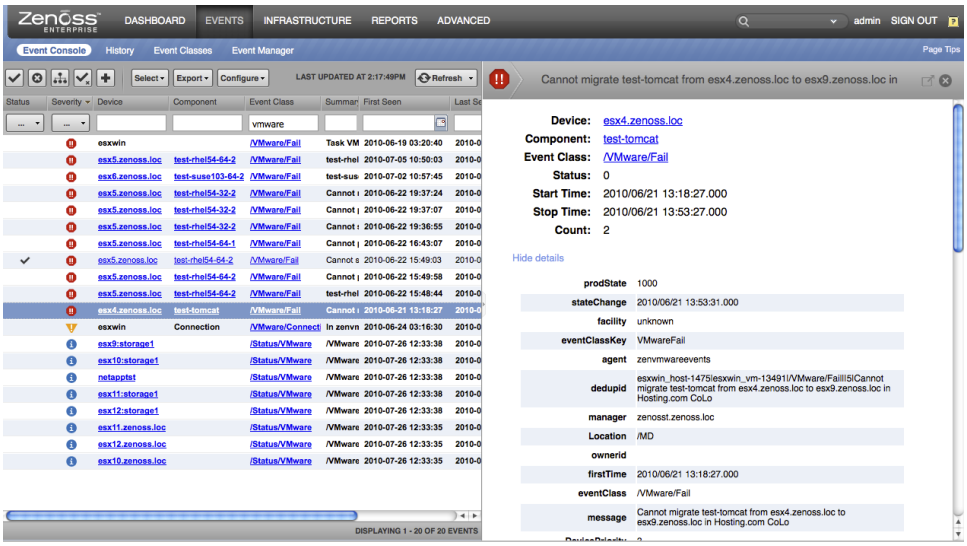
If a VMware host is disconnected, performance collection stops and the interface reflects its disconnected status.

Resource Manager maps the VMware event to the event class and assigns the event a severity level. The event class appears in the Event Class column.

To see detailed event information and the original VMware event type, double-click the event row.

The VMware event type is the value shown for eventGroup.

Figure 32: Event Details



Migration Events

When a VMotion guest migrates from one host to another, VMware records events to signal its progress. When a VmMigrated event occurs, it is duplicated to become two events, which are mapped to the `/VMware/Migration` event class in Resource Manager. One event contains the originating host as the device; the other lists the destination host as the device.

An event command (navigate to Events > Event Manager, and then select Commands in the left panel) reacts to these events by remodeling the two hosts and generating an updated view of the guests. The time required to produce updated guest lists (from the time migration completes) is between 30 seconds and four minutes.

Prerequisites

The VMware VI API is required. It is compatible with VMware Infrastructure 3 (including ESX Server 3.5, VirtualCenter Server 2.5, and ESX Server 3i) and vSphere 4 and 5 (including ESX 4, ESXi 4 and 5, vCenter Server 4 and 5). It is not explicitly compatible with ESX Server 3.0.x or VirtualCenter Server 2.0.x, or any previous versions.

Prerequisite	Restriction
Product	Resource Manager 4.x
Required ZenPacks	ZenPacks.zenoss.ZenVMware, ZenPacks.zenoss.StorageBase,

Prerequisite	Restriction
	ZenPacks.zenoss.DynamicView

Note If the time on the monitored VC/ESX server is too far form the time on the box where the `zenvmwareperf` daemon is running, the daemon will not collect any data.

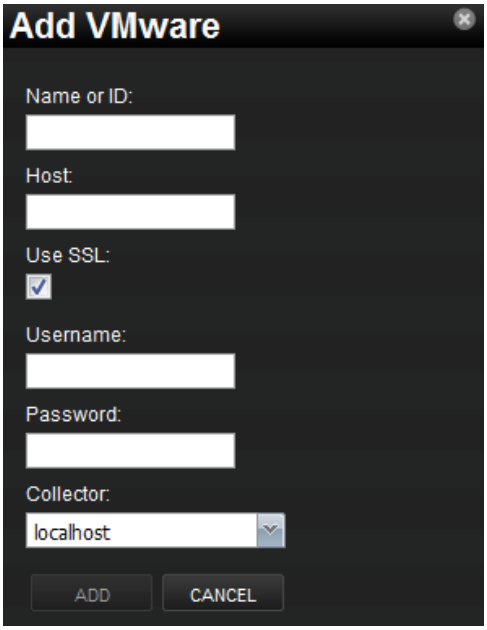
Enable Monitoring

Follow these steps to begin monitoring your VMware servers.

1 From Infrastructure > Devices, select Add VMware Infrastructure from the Add Device menu.

The Add VMware dialog appears.

Figure 33: Add VMware Infrastructure Dialog



2 Enter parameters to connect to the ESX server or vCenter Server that will provide monitoring capabilities.

- **Name or ID** -Enter a name for the infrastructure to be monitored.
- **Host** - Enter the hostname of the server providing the VI SDK connections. This can be an individual ESX server or the location of a vCenter Server instance.
- **Use SSL** - Select this option if the connection should be made by using SSL encryption.
- **Username** - Enter the user name used to authenticate.
- **Password** - Enter the password used to authenticate.
- **Collector** - Select the collector to use to retrieve information from the VI SDK endpoint.

3 Click **Add**.

Resource Manager begins modeling the VMware infrastructure. It places the information in the device hierarchy under `/Devices/VMware/ID`, where ID is the value of the ID field you entered during setup.

Note Do not model the same VMware infrastructure client with different names.

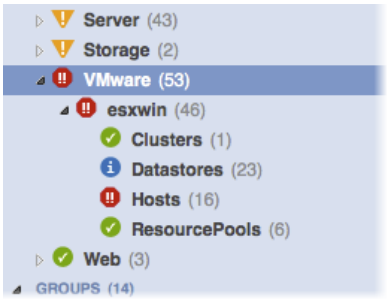
Viewing VMware Devices

Resource Manager represents these VMware entities as devices:

- Hosts (ESX servers)
- Resource Pools
- Data stores
- Clusters

Each of these categories is represented as a device class under the newly created organizer. For example, if the ID of an infrastructure is esxwin, then four device classes appear below /Devices/VMware/esxwin: Clusters, Datastores, Hosts, and ResourcePools.

Figure 34: VMware Device Classes



If the SDK endpoint is an individual ESX server, then the Clusters organizer will be empty. (A VMware cluster is a concept external to an individual host.)

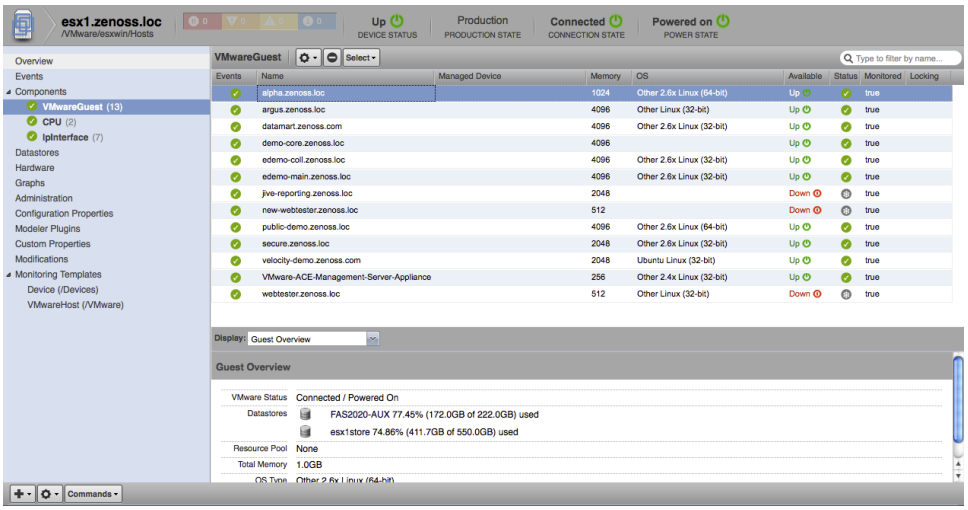
Viewing Guest Virtual Machines

To view guest VMs on an ESX server:

1 Navigate to a device in the Hosts class.
2 Select VMware Guest in the host's component tree (in the left panel).

The Virtual Guest Devices list appears.

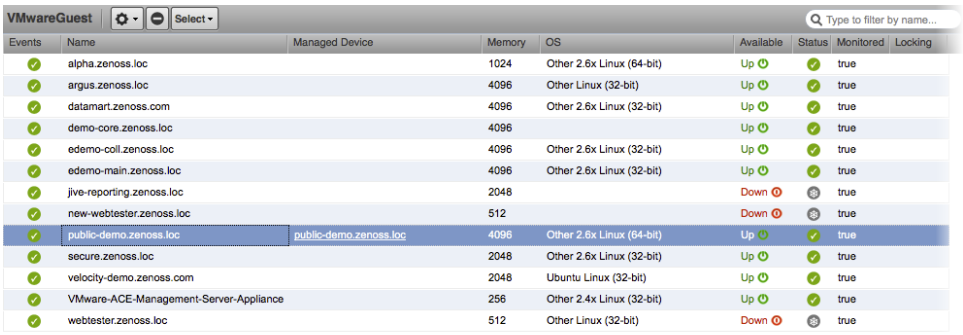
Figure 35: Virtual Guest Devices



In the list, the first column contains a link to the guest component, named the same name as the VM. (This is not necessarily the same as the VM hostname.) If the VM has been modeled elsewhere in Resource Manager, then a link to that device appears in the Managed Device column.

As shown in the previous figure, none of the VMs are being monitored in their "native" device classes. For example, the guest named "ldap test box" is a Linux VM with the hostname "public-demo.zenoss.loc." If you add that device to /Devices/Server/Linux, a link will appear.

Figure 36: Virtual Guest Devices - Managed Device



Events	Name	Managed Device	Memory	OS	Available	Status	Monitored	Locking
	alpha.zenoss.loc		1024	Other 2.6x Linux (64-bit)	Up		true	
	argus.zenoss.loc		4096	Other Linux (32-bit)	Up		true	
	datamart.zenoss.com		4096	Other 2.6x Linux (32-bit)	Up		true	
	demo-core.zenoss.loc		4096		Up		true	
	edemo-coll.zenoss.loc		4096	Other 2.6x Linux (32-bit)	Up		true	
	edemo-main.zenoss.loc		4096	Other 2.6x Linux (32-bit)	Up		true	
	jive-reporting.zenoss.loc		2048		Down		true	
	new-webtester.zenoss.loc		512		Down		true	
	public-demo.zenoss.loc	public-demo.zenoss.loc	4096	Other 2.6x Linux (64-bit)	Up		true	
	secure.zenoss.loc		2048	Other 2.6x Linux (32-bit)	Up		true	
	velocity-demo.zenoss.com		2048	Ubuntu Linux (32-bit)	Up		true	
	VMware-ACE-Management-Server-Appliance		256	Other 2.4x Linux (32-bit)	Up		true	
	webtester.zenoss.loc		512	Other Linux (32-bit)	Down		true	

Click the Name link to go to the Guest component status page, which shows the VM's relationships to other VMware entities, and provides access to VMware-specific metrics and events.

Click the managed device link to go to the Device status page, which contains information about the device as a separate Linux or Windows server. These two status pages link to each other.

Enabling Data Collection Using resxtop

Follow these steps to enable gathering of VMware host and guest statistics.

Gathering VMware Host Statistics

By default, data collection using `resxtop` statistics is disabled. To enable it:

1. From the Resource Manager interface, select Advanced, and then select Monitoring Templates.
2. Locate and select the VMwareHost_esxtop template.
3. For each of the data sources:

    a. Click the data source to open it.
    b. Select the Enabled option to enable data collection.
    c. Click **Save**.

    Data collection will begin shortly after update, followed by visible graph data.

    For information about the collected data, see Section 7, "Batch Mode," in the document titled "Interpreting esxtop Statistics" at the following location:

    http://communities.vmware.com/docs/DOC-9279

Gathering VMware Guest Statistics

By default, data collection using `resxtop` statistics is disabled. To enable it:

1. From the Resource Manager interface, select Advanced, and then select Monitoring Templates.

2 Locate and select the VMwareGuest_esxtop template.
3 For each of the data sources:

   a Click the data source to open it.
   b Select the Enabled option to enable data collection.
   c Click **Save**.

   Data collection will begin shortly after update, followed by visible graph data.

   For information about the collected data, see Section 7, "Batch Mode," in the document titled "Interpreting esxtop Statistics" at the following location:

   http://communities.vmware.com/docs/DOC-9279

Adding a Custom Metric

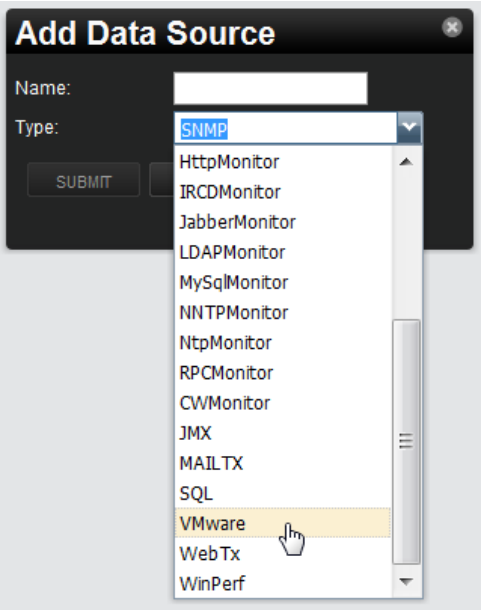
In Resource Manager, metric-bearing VMware entities (such as Hosts, Guests, and Clusters) have associated templates. These templates define which metrics are gathered. By default, only a subset is collected; however, you can add more by adding data sources to the templates. Once created, you can then create custom graphs from these data sources.

To create a custom data source:

1 Navigate to Advanced > Monitoring Templates and select the template to which you want to add the data source.
2 From the Data Sources area, click Add.

   The Add Data Source dialog appears.

Figure 37: Add Data Source



3 Enter a name and select the `VMware` data source from the list of options, and then click **Submit**.
4 Double-click the newly created data source to edit it. Enter or select values:

- **Event Key** - Not used.
- **Severity** - Not used.

- **Group**, **Counter**, and **Rollup Type** - VMware-specific data points are determined by this trio of strings. For information about each of these metrics, see the chapter titled "Performance Counters Reference" in the *VI SDK Programming Guide*.
- **Instance** - Certain metrics are further specified by an instance name. For example, the metric whose Group/Counter/Rollup Type triplet is Network/Network Data Receive Rate/average requires the name of the actual interface for full specification. In Resource Manager, this metric is represented by the data source nicRx on the template VMwareNic. The VMwareNic template is bound to the individual host interfaces, each of whose ID is the interface name. In this case, the instance name is ${here/instanceId}.

5 Click **Save** to save the new data source.

Moving VMware Devices Between Collectors

If you move a VMware device to a different collector, you must follow one of these procedures to force the changes to take effect:

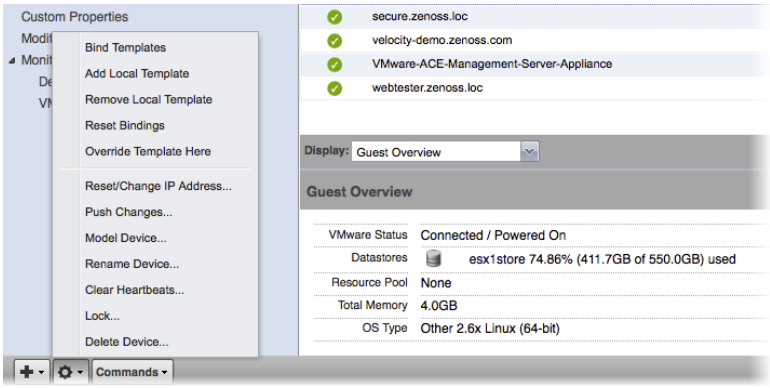
- Restart the collector daemons. To do this, go to Advanced > Settings, select Daemons in the left panel, and then click **Restart** in the row for each of these daemons:
  - zenvmwaremodeler
  - zenvmwareperf
  - zenvmwareevents

  Note Alternatively, as user zenoss, enter the following commands to stop and then restart these Resource Manager daemons:

```
zenvmwaremodeler restart
zenvmwareperf restart
zenvmwareevents restart
```

  OR
- Navigate to the page for the organizer that represents the VMware endpoint (for example, `Devices/VMware`, myEndpoint), and then select Push Changes from the Action menu.

Figure 38: Push Changes



Daemons

Type	Name
Modeler	`zenvmwaremodeler`

Type	Name
Performance Collector	`zenvmwareperf`
Event Collector	`zenvmwareevents`

Tuning Options

These collector daemons offer options for tuning performance. Use them to control data amounts and the rate at which data comes back to be modified.

- `zenvmwareperf`

Table 61: Daemons

Option	Description
`--callChunkSize=`*Value*	Specifies the number of performance requests to submit at the same time.
`--callChunkSleep=`*Value*	Specifies the time to sleep, in seconds, between performance requests.

- `zenvmwareevents`

Table 62: Daemons

Option	Description
`--eventChunkSize=`*Value*	Specifies the number of events to gather at one time.
`--eventChunkSleep=`*Value*	Specifies the time to sleep, in seconds, between event requests.

(ZenWebTx) Web-Based Synthetic Transactions 66

The ZenPacks.zenoss.ZenWebTx ZenPack adds the `zenwebtx` daemon, which enables availability and performance monitoring of web sites through synthetic HTTP transactions.

Synthetic transactions perform some of the same activities performed by your user community. You create one or more tests that mimic user actions in a Web browser. Resource Manager then performs these tests periodically, creating events when a test fails or exceeds a time threshold.

Additionally, Resource Manager can record data for each test run, such as:

- Time required for the test to execute
- Time taken for any portion of the test to complete
- Values extracted from Web pages during the test

This ZenPack uses a scripting language called twill to describe the steps of a test. These steps include actions such as:

- Clicking a link
- Completing form fields
- Assertions, which check for the presence or absence of text on a page. In addition, you can extract data from the Web page and record the numeric values that are a part of these patterns
- Descriptions of data to collect during the test

You can write twill commands manually. You also can use a Firefox add-on called TestGen4Web to record a browser session that this ZenPack then translates into twill commands. The `zenwebtx` daemon processes the twill commands periodically, recording data and creating events as appropriate.

Related Links

twill command reference on page 247

Data Points

Data produced by any Resource Manager data source are called data points. `WebTx` data sources contain two default data points:

- **totalTime** – Number of seconds taken to complete the entire transaction.
- **success** – Returns 1 (success) or 0 (failure), depending on whether or not the transaction succeeded.

You can create other data points by using the extract and printTimer twill commands, which output data values when the twill commands are run. You must create new data points with the same name you used in those commands to bring that data into Resource Manager.

The `zenwebtx` daemon supports XPath queries to extract data from XML documents.

Event Generation

There are several situations in which this ZenPack creates events in Resource Manager. These events use the component and event class specified on the Data Source tab. These situations are:

- The `zenwebtx` daemon is unable to retrieve a page during the transaction.
- One of the twill commands fails, such as finding text that does not exist or following a link that does not exist.
- The timeout (specified on the Data Source tab) is exceeded.
- A threshold defined for one of the data points in this data source is exceeded. Thresholds are defined in the monitoring template that contains the data source.

Enable Monitoring

1 To create a `WebTx` data source:

1 From the data sources area, click Add Data Source.

2 In the Create Data Source dialog, enter the name of the new data source, and then select the data source type `WebTx`.

3 Click **Submit**.

4 Select the data source to edit it. Enter information or make selections to specify how and when this data source's Web transactions are performed, and which data should be collected:

Table 63: WebTx Data Source Options

Option	Description
Name	Displays the name of the data source that you specified in the Create Data Source dialog. This name is used in thresholds and graph definitions to refer to the data collected by this data source.
Source Type	Set to `WebTx`, indicating that this is a synthetic Web transaction data source. You cannot edit this selection.
Enabled	Set to True (the default) to collect information from this data source. You may want to set this value to False to disable data sources when developing the data source, or when making changes to the Web application being tested.
Component	Any time the Web transaction fails, Resource Manager generates an event. Use this field to set the Component field of the generated event.
Event Class	Select the event class of the event generated by this data source. Normally, this is set to /Status/Web (according to the value set on the data source).
Timeout	Specify the number of seconds that zenwebtx will attempt to execute this data source's commands before it generates an error event.
Cycle Time	Specify the number of seconds that zenwebtx will wait between the start of one test run and the start of the next.
User Agent	Specify the text that zenwebtx will present to target Web sites to identify itself.

5 Click **Save** to save the specified settings.

6 Select **Script**. From here, you will specify the details of the transaction. Information here also helps you debug twill commands when setting up the data source.

7 Note If you provide values for Initial User, Initial Password, and Initial Authentication Realm, Resource Manager will use these credentials before accessing the URL specified for Initial URL. All three (Initial User, Initial Password, and Initial Authentication Realm) must be present; otherwise, the values are ignored.

Enter information or make selections:

Table 64: WebTx Script Settings

Option	Description
Initial URL	Specify the URL of the page where the transaction will start. This field frequently contains a TALES expression to refer to a device's ID or IP address, such as `http://${dev/id}` or `http://${dev/manageIp}`.
Initial User	Specify the user name for authentication.
Initial Password	Specify the user password for authentication.
Initial Authentication Realm	Specify the basic HTTP authentication realm.
TestDevice	Use this field to test and debug twill commands. Enter the ID of a device, and then click **Test Twill Commands** to execute the twill commands against the device. If you do not specify a device, then Resource Manager will select a device for you.
Upload Recording	Upload a Web session recording generated by the Firefox TestGen4Web add-on. Enter or browse to the recording location. If you specify a file here, and then click **Save**, Resource Manager translates the file to twill commands and replaces the contents of the Twill Commands field with the newly translated commands.
Twill Commands	Specify the number of seconds that zenwebtx will wait between the start of one test run and the start of the next. Enter twill commands that Resource Manager will execute to produce values and events for the data source. If you select this action, then the current contents of the Twill Commands field is completely replaced. Resource Manager does not save the replaced information.

8 Click **Save** to save the data source.

Creating twill Commands

This ZenPack uses twill to specify the steps of a Web test. Each `WebTx` data source has a field that contains the twill commands that describe a Web transaction. You can create this list of twill commands manually, or you can record a session in a browser and use that as the basis for your data source.

Some twill commands specify an action, such as following a specific link on a page or entering data in a form field. Other twill commands specify a test, such as searching for specific text on a page or making sure the title does not contain specific text.

Creating twill Commands from TestGen4Web

The *TestGen4Web Firefox add-on* allows you to record browser sessions. This ZenPack can take these sessions and convert them to twill, creating a starting point for developing WebTx data sources.

1 Follow these general steps to record and convert a TestGen4Web session:
1 From the TestGen4Web toolbar in Firefox, use the **Record** and **Stop** buttons to record a session.
2 Use the **Save** button in the toolbar to save the session to a file.
3 From the Script page of a WebTx data source in Resource Manager, browse to and select your saved session.
4 Click **Save** to convert the TestGen4Web session to twill. The newly converted commands appear in the Twill Commands field on the page, replacing any previous twill commands in that area.

Creating twill Commands Manually

Even if you use TestGen4Web to initially create twill commands, you will frequently want to edit these commands manually to add data points or additional content checks. The Test Twill Commands button on the Script page is helpful when testing twill commands as you create or edit them.

You also can execute twill commands interactively by using the `twill-sh` program from the command line. This program lets you enter commands one at a time and then inspect the pages that come back.

Log in as user zenoss.

Define the needed PATH variables, and start the twill shell.

```
webtx=$(zenpack --list | grep ZenWebTx | \
  sed -e 's|^.*ZenPacks/||' -e 's|)||')
webtx_path=$ZENHOME/ZenPacks/${webtx}/ZenPacks/zenoss/ZenWebTx
export PYTHONPATH="${webtx_path}/lib:${PYTHONPATH}"
export PATH="${webtx_path}/bin:${PATH}"
twill-sh
```

Within `twill-sh`, use the help command to list available commands and see a command descriptions. Of particular interest are these commands:

- `showforms` – Lists the forms on the page and the fields within each.
- `showlinks` – Lists the links on the page.
- `show` – Lists the source HTML from the page.
- `exit` – Quits the `twill-sh` program.

Often the most convenient way to use `twill-sh` is to create a text file that contains your twill commands. You can then specify that file on the command line when you invoke `twill-sh`. This lets you analyze problems that occur.

Invoke `twill-sh` with a text file as such:

```
twill-sh -i myTwillCommands.txt
```

The `-i` option instructs `twill-sh` to stay in the twill shell rather than exiting when it finishes running the commands in the `myTwillCommands.txt` file.

Monitoring through Proxy Servers

The `zenwebtx` daemon can access Web servers through HTTP proxy servers and non-authenticating HTTPS proxy servers.

1 To configure the `zenwebtx` daemon to use a proxy, you must define the *http_proxy* and *https_proxy* environment variables.

1 Open the `~zenoss/.bashrc` file.

2 Add the following lines:

```
export http_proxy=http://Address:Port/
export https_proxy=http://Address:Port/
```

where `Address` is the address of your HTTP or HTTPS proxy server, and `Port` is the port on which your proxy server listens.

Example Proxy Setup

HTTP and HTTPS proxies frequently listen on port 3128. If your proxy server is "my.proxyserver.loc" and it uses port 3128, then add these two lines to the `~zenoss/.bashrc` file:

```
export http_proxy=http://my.proxyserver.loc:3128/
export https_proxy=http://my.proxyserver.loc:3128/
```

Testing the Proxy Setup

You can test the proxy setup by using `twill-sh`, the twill interpreter shell.

1 After setting up the proxy information in the `~zenoss/.bashrc` file, follow these steps to test your setup:

1 Make sure *http_proxy* and *https_proxy* are defined in your current shell:

```
$ source ~zenoss/.bashrc
```

2 Launch the twill shell:

```
webtx=$(zenpack --list | grep ZenWebTx | \
  sed -e 's|^.*ZenPacks/||' -e 's|)||')
webtx_path=$ZENHOME/ZenPacks/${webtx}/ZenPacks/zenoss/ZenWebTx
export PYTHONPATH="${webtx_path}/lib:${PYTHONPATH}"
export PATH="${webtx_path}/bin:${PATH}"
twill-sh
```

3 Try to retrieve a URL through HTTP or HTTPS. For example, to retrieve the Resource Manager home page, enter:

```
go http://www.zenoss.com
```

You should see a message similar to this:

```
current page: http://www.zenoss.com
```

If an error message appears, then your proxy may not be correctly configured in the `~zenoss/.bashrc` file.

4 Exit the twill shell:

```
exit
```

twill command reference

Browsing

- `go` <*URL*> - Visit the given URL.
- `back` - Return to the previous URL.
- `reload` - Reload the current URL.
- `follow` <*link name*> - Follow a link on the current page.

Assertions

- `code` <*code*> - Assert that the last page loaded had this HTTP status. For example, ``code 200`` asserts that the page loaded correctly.
- `find` <*regexp*> - Assert that the page contains this regular expression.
- `notfind` <*regexp*> - Assert that the page does not contain this regular expression.
- `url` <*regexp*> - Assert that the current URL matches the given regexp.
- `title` <*regexp*> - Assert that the title of this page matches this regular expression.

Display

- `echo` <string> - Echo the string to the screen.
- `redirect_output` <filename> - Append all Twill output to the given file.
- `reset_output` - Display all output to the screen.
- `save_html` [<filename>] - Save the current page's HTML to a file. If no filename is given, derive the filename from the URL.
- `show` - Show the current page's HTML.
- `showlinks` - Show all of the links on the current page.
- `showforms` - Show all of the forms on the current page.
- `showhistory` - Show the browser history.

Forms

- `submit *[<n>]*` - Click the nth submit button, if given; otherwise, submit via the last submission button clicked. If nothing is clicked, then use the first submit button on the form. See the section titled Details on Form Handling for more information.
- `formvalue` <formnum> <fieldname> <value> - Set the given field in the given form to the given value. For read-only form widgets and controls, the click may be recorded for use by submit, but the value is not changed unless the `config` command has changed the default behavior. See `config` and the section titled "Details on Form Handling" for more information on the `formvalue` command.

  For list widgets, you can use one of the following commands to select or de-select a particular value. To select a value, enter the command in this format:
  `formvalue <formnum> <fieldname> +value`

  To de-select a value:
  `formvalue <formnum> <fieldname> -value`
- `fv` - Abbreviation for the formvalue command.
- `formaction` <formnum> <action> - Change the form action URL to the given URL.
- `fa` - abbreviation for the fa command.
- `formclear` - Clear all values in the form.

- `formfile` <formspec> <fieldspec> <filename> [<content_type>]* - attach a file to a file upload button by filename.

Cookies

- `save_cookies` <filename> - Save the current cookie jar to a file.
- `load_cookies` <filename> - Replace the current cookie jar with the specified file contents.
- `clear_cookies` - Clear all of the current cookies.
- `show_cookies` - show all of the current cookies. Sometimes useful for debugging.

Debugging

`debug` <what> <level> - Turn on or off debugging/tracing for various functions.

Enter the command in the form:
`debug <what> <level>`

where <what> is one of these options:

- `HTTP` - Show HTTP headers.
- `equiv-refresh` - Test HTTP EQUIV-REFRESH headers.
- `twill` - Show twill commands.

and <level> is 0 (for off) or 1 (for on).

Other Commands

- `tidy_ok` - Check to see if the `tidy` command runs on this page without any errors or warnings.
- `exit` *[<code>]* - Exit with the given integer code, if specified. The value of <code> defaults to 0.
- `run` <command> - Execute the specified Python command.
- `run file` <file1> [<file2> ...]* - Execute the specified files.
- `agent` - Set the browser's "User-agent" string.
- `sleep` [<seconds>] - sleep the given number of seconds. Defaults to 1 second.
- `reset_browser` - Reset the browser.
- `extend_with` <module> - Import commands from the specified Python module. This acts like ``from <module> import *`` does in Python.

  For example, a function ``fun`` in ``ext module`` would be available as ``fun``. See *examples/extend_example.py* for an example.
- `add_auth` <realm> <uri> <user> <password> - Add HTTP Basic Authentication information for the given realm/URL combination.

  For example, “add_auth IdyllStuff http://www.idyll.org/ titus test” tells twill that a request from the authentication realm "IdyllStuff" under http://www.idyll.org/ should be answered with username 'titus', password 'test'. If the 'with_default_realm' option is set to True, ignore 'realm'.
- `config` [<key> [<value>]] - Show/set configuration options.
- `add_extra_headers` <name> <value> - Add an extra HTTP header to each HTTP request.
- `show_extra_headers` - Show the headers being added to each HTTP request.
- `clear_extra_headers` - Clear the headers being added to each HTTP request.

Details on Form Handling

The `formvalue` (or `fv`) and `submit` commands rely on a certain amount of implicit cleverness to do their work. In odd situations, it is difficult to determine which form field `formvalue` will choose based on your field name, or which form and field `submit` is going to "click" on.

Example 1

Following is the pseudocode for how `formvalue` and submit determine which form to use (function 'twill.commands.browser.get_form')::

for each form on page:

if supplied regexp pattern matches the form name, select

if no form name, try converting to an integer N & using N-1 as

an index into the list or forms on the page (for example, form 1 is

the first form on the page).

Example 2

Following is the pseudocode for how `formvalue` and `submit` determine which form field to use (function `twill.commands.browser.get_form_field`)::

search current form for control name with exact match to fieldname;

if single (unique) match, select.

if no match, convert fieldname into a number and use as an index, if

possible.

if no match, search current form for control name with regexp match to fieldname;

if single (unique) match, select.

if *still* no match, look for exact matches to submit-button values.

if single (unique) match, select.

Example 3

Following is the pseudocode for `submit`::

if a form was _not_ previously selected by `formvalue`:

if there is only one form on the page, select it.

otherwise, fail.

if a field is not explicitly named:

if a submit button was "clicked" with `formvalue`, use it.

otherwise, use the first submit button on the form, if any.

otherwise:

find the field using the same rules as `formvalue`

finally, if a button has been picked, submit using it;

otherwise, submit without using a button

twill Extensions

The ZenPacks.zenoss.ZenWebTx ZenPack extends the standard twill vocabulary with the commands described in this section.

twilltiming

twilltiming sets timers in a set of twill commands. If you then define a data point for this timer, you can graph and set thresholds on this timer value.

Use the following command to start a new timer:
```
startTimer myTimerName
```

and then, to output the value:
```
printTimer myTimerName
```

Timer values should be output only once. So, to output the time from the start of the script to more than one point in the script, you must use more than one timer. For example:

```
startTimer wwwZenossCom
startTimer bothPages
go http://www.zenoss.com
printTimer wwwZenossCom
startTimer communityPage
follow "Community"
printTimer communityPage
printTimer bothPages
```

To use these timers in Resource Manager, create data points with the same name as the timers. In this example you could create data points named wwwZenossCom, communityPage, and bothPages. You can then use these data points in Resource Manager thresholds and graph definitions.

twillextract

twillextract extracts numeric values from Web pages during the transaction. To use twillextract, use the following command to match the given regular expression to the current page:
```
extract <dataName> <regularExpression>
```

The value 1 or 0 is assigned to dataName depending on whether the regular expression matched or not.

Additionally, you can use Python's regular expression substring-matching syntax to extract substrings of the matched text. For example, http://www.zenoss.com contains a copyright notice near the bottom that looks like "Copyright (c) 2005-2011 Zenoss, Inc." The following twill commands use a regular expression to grab the second year from that notice:

```
go http://www.zenoss.com
extract copyright "(?P<firstYear>[0-9]*)-(?P<secondYear>[0-9]*) Zenoss,
 Inc."
```

`(?P<name>.....)` is Python syntax for naming that particular part of the regular expression. The value extracted from that part of the matching text is given the name from the extract command, then a dash, then the name from the sub-pattern. In this example, copyright gets a value of 1 or 0 depending on whether the pattern was found on the page or not, and copyright-firstYear and copyright-secondYear get the values extracted from the matched text. To use these values in Resource Manager you must create data points in the WebTx data source with the same name as those you used in the extract command. In this case you would create data points named copyright, copyright-firstYear and copyright-secondYear. You can then create graph definitions and thresholds for these data points.

twillxpathextract

Resource Manager uses the twillxpathextract command to extract numeric values from XML documents. To use twillxpathextract, add the following command to match and extract data using the given XPath expression:
`xpathextract <dataName> <xpath>`

where `xpathextract` is the command name, <dataName> is the name of the data point to which the value will map, and <xpath> is the Xpath expression used to retrieve the data.

When applied to an XML document, XPath expressions must return a numeric value. This value is then assigned to the dataName data point.

ignorescripts

ignorescripts strips javascript from visited pages before they are processed by twill. Although twill ignores script tags, it is possible for scripts to include strings that twill will interpret as HTML tags. Including the command extend_with ignorescripts near the top of your twill commands will cause all script tags to be stripped, thereby avoiding this issue.

Administrating ZenPacks

Installing and upgrading ZenPacks

The ZenPack installation and update procedure requires stopping and starting Resource Manager.

1 Log in to the Resource Manager master host as `zenoss`.
2 Download the ZenPack or ZenPacks you wish to install or upgrade from the *Zenoss Support* site.
Contact your Zenoss representative for login credentials.
3 Stop Resource Manager.

- *Stopping Resource Manager without remote collector or hub hosts* on page 253
- *Stopping Resource Manager with remote collector or hub hosts* on page 253

At the end of both procedures, you are logged in to the Resource Manager master host as `zenoss`.
4 Start the event and catalog servers.
```
zeneventserver start; zencatalogservice start
```
5 Install the new ZenPacks.
```
zenpack --install ZenPack.Name-Version.egg
```
6 If a ZenPack introduces a new daemon, and you are using `$ZENHOME/etc/daemons.txt`, add the new daemon to the file.
7 Start Resource Manager.
```
zenoss start
```
8 Update remote collectors, if deployed. Repeat the following steps for each remote collector.
a Log in to the Resource Manager user interface as a user with ZenManager or Manager privileges.
b Click **ADVANCED**, and then **Collectors**.
c Display the collector's overview page.
d In the **Performance Collector Configuration** panel, select **Update Collector...** from the **Action** menu.
e In the **Update Collector** dialog, click **OK**.
9 Update remote hubs, if deployed. Repeat the following steps for each remote hub.
a Log in to the Resource Manager user interface as a user with ZenManager or Manager privileges.
b Click **ADVANCED**, and then **Collectors**.
c Display the hub's overview page.
d In the **Hub Configuration** panel, select **Update Hub...** from the **Action** menu.
e In the **Update Hub** dialog, click **OK**.

Removing a ZenPack

The ZenPack removal procedure requires stopping and starting Resource Manager.

Note Removing a ZenPack can have unexpected consequences.

- Removing a ZenPack removes all objects provided by the ZenPack, as well as all objects that depend on code provided by the ZenPack. Review the details page of the ZenPack to remove, before removing it.
- Removing a ZenPack that installs a device class removes the device class, any contained device classes, and all devices in that class.

To avoid negative consequences, Zenoss recommends the following steps before removing a ZenPack.

- Delete data sources provided by the ZenPack to remove, if any.
- Back up Resource Manager data with the `zenbackup` command.

---

1. Log in to the Resource Manager master host as `zenoss`.
2. From the list of installed ZenPacks, identify the name of the ZenPack to remove.
   ```
   zenpack --list
   ```
   The output includes the full name of the ZenPack; a blank space; and the ZenPack location, in parentheses.
3. Stop Resource Manager.
   - *Stopping Resource Manager without remote collector or hub hosts* on page 253
   - *Stopping Resource Manager with remote collector or hub hosts* on page 253

   At the end of both procedures, you are logged in to the Resource Manager master host as `zenoss`.
4. Start the event and catalog servers.
   ```
   zeneventserver start; zencatalogservice start
   ```
5. Remove the ZenPack. Replace *Name* with the full name of the ZenPack to remove.
   ```
   zenpack --remove=Name
   ```
6. If the removed ZenPack provides a daemon, and you are using `$ZENHOME/etc/daemons.txt`, delete the daemon from the file.
7. Start Resource Manager.
   ```
   zenoss start
   ```
8. Update remote collectors, if deployed. Repeat the following steps for each remote collector.
   a. Log in to the Resource Manager user interface as a user with ZenManager or Manager privileges.
   b. Click **ADVANCED**, and then **Collectors**.
   c. Display the collector's overview page.
   d. In the **Performance Collector Configuration** panel, select **Update Collector...** from the **Action** menu.
   e. In the **Update Collector** dialog, click **OK**.
9. Update remote hubs, if deployed. Repeat the following steps for each remote hub.
   a. Log in to the Resource Manager user interface as a user with ZenManager or Manager privileges.
   b. Click **ADVANCED**, and then **Collectors**.
   c. Display the hub's overview page.
   d. In the **Hub Configuration** panel, select **Update Hub...** from the **Action** menu.
   e. In the **Update Hub** dialog, click **OK**.

Stopping Resource Manager without remote collector or hub hosts

When a Resource Manager deployment does not include remote hub or collector hosts, stopping all daemons is simple.

1. Log in to the Resource Manager master host as `zenoss`.
2. Stop all Resource Manager daemons.
   ```
   zenoss stop
   ```
   Occasionally, the `stop` command does not terminate all of the daemons.
3. Check for daemons that are not stopped.
   ```
   pgrep -fl ${ZENHOME}
   ```
   - If the command returns no result, Resource Manager is stopped.
   - If the command returns a result, stop the remaining daemons.
     ```
     pkill -f ${ZENHOME}
     ```

Stopping Resource Manager with remote collector or hub hosts

Stop all daemons on remote Resource Manager collector or hub hosts before stopping them on the master host.

1. Log in to the Resource Manager master host as `zenoss`.
2. Stop the `zenwebserver` daemon.

```
zenwebserver stop
```

3. Stop Resource Manager daemons on all collector hosts.
   a. Log in to each collector host as `zenoss`.

```
ssh zenoss@Remote-Collector-Host
```

   b. Stop all Resource Manager daemons.

```
zenoss stop
```

   c. Check for daemons that are not stopped.

```
pgrep -fl ${ZENHOME}
```

   - If the command returns no result, Resource Manager is stopped.
   - If the command returns a result, stop the remaining daemons.

```
pkill -f ${ZENHOME}
```

4. Stop Resource Manager daemons on all hub hosts.
   a. Log in to each hub host as `zenoss`.

```
ssh zenoss@Remote-Hub-Host
```

   b. Stop all Resource Manager daemons.

```
zenoss stop
```

   c. Check for daemons that are not stopped.

```
pgrep -fl ${ZENHOME}
```

   - If the command returns no result, Resource Manager is stopped.
   - If the command returns a result, stop the remaining daemons.

```
pkill -f ${ZENHOME}
```

5. Stop all Resource Manager daemons on the master host.
   a. Log in to the Resource Manager master host as `zenoss`.
   b. Stop all Resource Manager daemons.

```
zenoss stop
```

   c. Check for daemons that are not stopped.

```
pgrep -fl ${ZENHOME}
```

   - If the command returns no result, Resource Manager is stopped.

- If the command returns a result, stop the remaining daemons.

```
pkill -f ${ZENHOME}
```

Resource Manager daemons

Resource Manager daemons

The daemons that are always part of Resource Manager.

zenactiond

Runs background jobs such as email notification, database aging, and maintenance window processing.

zencommand

Collects performance data from devices by running commands and parsing the output.

zeneventd

Performs event mappings, transformations, and other data-driven event processing tasks. Forwards processed events to **zeneventserver**.

zeneventserver

Stores and retrieves events from its database, `zenoss_zep`. Performs deduplication and clearing; trigger analysis and trigger signal queuing; indexing and storage of the events into Zenoss DataStore; and fanout queuing.

zenhub

zenhubworker

Provides an intermediate connection between collectors and the modeling and event databases. May be run on the master server, on remote servers, or both.

zenjobs

Performs tasks in its queue as scheduled. Large tasks, such as adding a device, may be queued instead of performed immediately. Since version 4.2.2, **zenjobs** is based on Celery and uses RabbitMQ for job queuing.

zenmodeler

Collects device characteristics through SNMP, SSH, Telnet, and WMI, at scheduled intervals. (Except VMware devices; see **zenvmwaremodeler**.)

zenperfsnmp

Collects status and performance data from devices through SNMP. Monitoring templates define the data to collect.

zenping

Checks device status and network health with TCP packets. Since version 4.1.1, zenping uses Nmap to build a ping tree and perform Layer 3 event suppression.

zenprocess

Checks the status of processes on modeled devices, at scheduled intervals.

zenrrdcached

Provides a queue of writes for RRD data files, to improve the efficiency of collector hosts.

zenstatus

Checks the status of Resource Manager daemons on `localhost`. Runs on the Resource Manager master host and all remote hub and collector hosts.

zensyslog

Collects and classifies `syslog` events.

zentrap

Collects and parses SNMP traps, resolves OIDs into MIB names, and then forwards the traps to **zenhub** for additional processing.

zredis

Provides a shared repository for all **zenping** daemons, and facilitates correlation of "ping down" events.

Daemons from ZenPacks

The daemons that are installed when specific ZenPacks are installed.

Daemon	Description	Source ZenPack
zencatalogservice	Creates and maintains a full-text index of the event server database, which improves search performance in the console interface.	**ZenPacks.zenoss.CatalogService**
zeneventlog	Collects events from Windows Management Instrumentation (WMI) event logs.	*WindowsMonitor (Microsoft Windows)* on page 197
zenjmx	Enables monitoring of Java applications by communicating with remote Java Management Extensions (JMX) agents.	*(ZenJMX) Java Management Extensions* on page 212
zenjserver	Enables the dynamic view of organizers to include objects that can impact the status of organizers, such as other organizers and devices. Renders the Java-based Dynamic Views found in the console interface.	*(DynamicView) Dynamic Service View* on page 71
zenmailtx	Enables round-trip monitoring of email messages, to monitor email servers.	*(ZenMailTx) Mail Transactions* on page 225
zenpython	Runs Python monitoring code supplied by other ZenPacks.	ZenPacks.zenoss.PythonCollector
zentune	Performs analyses of Resource Manager components and services.	*(AutoTune) ZenTune* on page 22
zenucsevents	Collects events from Cisco Unified Computing System (UCS) event logs.	*(CiscoUCS) Cisco UCS* on page 46
zenvcloud	Monitors VMware vCloud environments.	*(vCloud) VMware vCloud* on page 173
zenvmwareevents	Collects VMware event log events.	*(ZenVMware) VMware vSphere* on page 233
zenvmwaremodeler	Collects device characteristics from VMware endpoints.	*(ZenVMware) VMware vSphere* on page 233
zenvmwareperf	Collects performance data from VMware endpoints.	*(ZenVMware) VMware vSphere* on page 233
zenvsphere	Collects performance data through a VMware vCenter server.	*(vSphere) VMware vSphere* on page 177
zenwebserver	(Not a daemon.) A script to deploy and manage multiple Zope instances using Nginx.	*(WebScale) WebScale* on page 189
zenwebtx	Checks the availability and performance of Web sites with synthetic transactions. Employs twill scripts to perform transactions that mimic the activities of a site's users.	*(ZenWebTx) Web-Based Synthetic Transactions* on page 241
zenwin	Checks the status of Windows services.	*WindowsMonitor (Microsoft Windows)* on page 197

Daemon	Description	Source ZenPack
zenwinperf	Collects status and performance data from devices through Windows Performance Monitor. Monitoring templates define the data to collect.	*WindowsMonitor (Microsoft Windows)* on page 197