EXTENDED MONITORING

Zenoss, Inc.

www.zenoss.com

Zenoss Extended Monitoring

The Zenoss logo is a registered trademark of Zenoss, Inc. Zenoss and Open Enterprise Management are trademarks of Zenoss, Inc. in the U.S. and other countries.

Amazon Web Services, AWS, Amazon Elastic Compute Cloud, and Amazon EC2 are trademarks of Amazon.com, Inc. or its affiliates in the United States and/or other countries.

Flash is a registered trademark of Adobe Systems Incorporated.

Oracle, the Oracle logo, and Java are registered trademarks of the Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Linux is a registered trademark of Linus Torvalds.

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

Sybase is a registered trademark of Sybase, Inc.

Tomcat is a trademark of the Apache Software Foundation.

VMware is registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions.

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

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

Part Number: 07-102010-3.0-v04

Chapter 1. ZenPacks

1.1. About ZenPacks

ZenPacks extend and modify the system to add new functionality. This can be as simple as adding new device classes or monitoring templates, or as complex as extending the data model and providing new collection daemons.

You can use ZenPacks to add:

- Monitoring templates
- Data sources
- Graphs
- Event classes
- Event and user commands
- Reports
- Model extensions
- Product definitions

Simple ZenPacks can be created completely within the user interface. More complex ZenPacks require development of scripts or daemons, using Python or another programming language.

ZenPacks can be distributed for installation on other Zenoss systems.

1.1.1. Provided ZenPacks

A range of provided ZenPacks add and extend system functionality. These ZenPacks are grouped as Core ZenPacks (available to all users) and Enterprise ZenPacks, which are available only to Zenoss Enterprise implementations.

The guide titled Zenoss Extended Monitoring provides detailed descriptions, installation information, and configuration details for Core and Enterprise ZenPacks.

The following sections provide information and procedures to help you:

- Install ZenPacks
- Create ZenPacks
- Package and distribute ZenPacks
- Remove ZenPacks

1.2. Installing ZenPacks

ZenPacks are distributed as `.egg` files. You can install ZenPacks from the command line on the Zenoss server, or from the user interface.

1.2.1. Installing from the Command Line

Use these commands to install a ZenPack file and then restart the system:

```
zenpack --install <filename>
```

```
zenoss restart
```

If you have the source code for the ZenPack you can install directly from that rather than from an `.egg` file. The command is the same; however, you must specify the directory containing the source code. This copies the source code to `$ZENHOME/ZenPacks`:

```
zenpack --install <directoryname>
zenoss restart
```

If you are developing a ZenPack, you should maintain your source code outside of `$ZENHOME/ZenPacks` for two reasons:

- if you issue a **zenpack --remove** command it will delete your code from that location and you will lose your files unless you have them backed up elsewhere.
- if you are maintaining your source code in a version control system it is frequently more convenient to have the files reside elsewhere on the file system.

Using the **--link** option, you can install the ZenPack but have the system use your code from its current location. Instead of installing your code in `$ZENHOME/ZenPacks`, the system will create a link in that location that points to your source code directory.

```
zenpack --link --install <directoryname>
zenoss restart
```

1.2.2. Installing from the User Interface

To upload and install a ZenPack `.egg` file from the user interface:

1. From the navigation bar, select Advanced > Settings.
2. In the left panel, select ZenPacks.
3. From ⚙▾ (Action menu), select Install ZenPack.

   The Install ZenPack dialog appears.
4. Browse to and select the `.egg` file you want to install, and then click **OK**.

   The file is uploaded to the Zenoss server and installed.

Note

After installing the ZenPack, you should restart the system.

1.2.3. Installing All Core ZenPacks from RPM

The core ZenPacks, along with third party ZenPacks, are available for download individually from:

http://community.zenoss.org/community/zenpacks

At that location is a link to download an RPM that includes the most popular core ZenPacks. To install via the core ZenPacks RPM follow these steps:

1. Download the appropriate file from the ZenPacks download area specific to your version.
2. Make sure ZEO is running (as the zenoss user):

   ```
   zeoctl start
   ```
3. Install the rpm (as root user):

```
rpm -ihv <rpm file>
```

4. Restart Zope and ZenHub:

```
zopectl restart
zenhub restart
```

1.2.4. Viewing Loaded ZenPacks

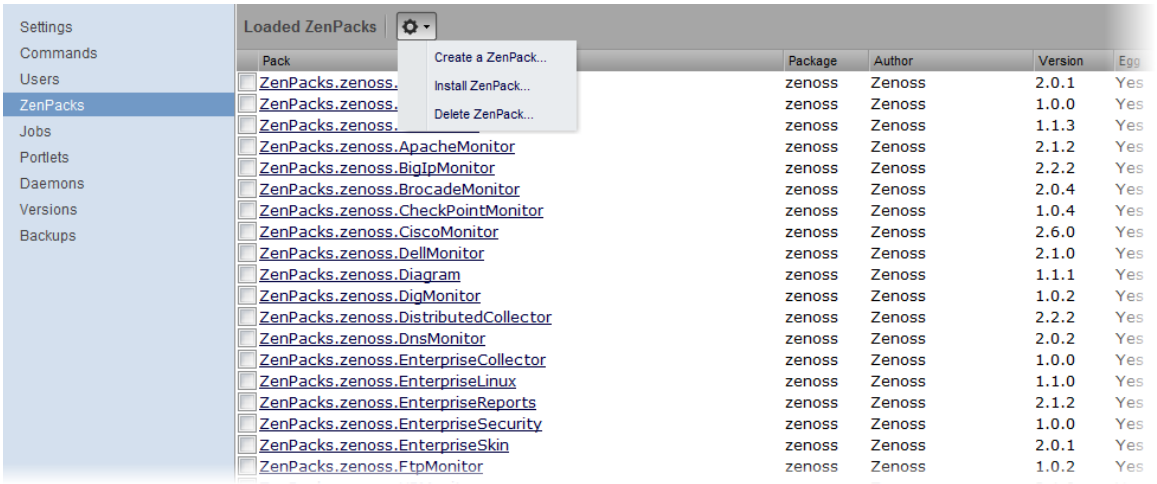
To see which ZenPacks are loaded on your system:

1. From the navigation bar, select Advanced.

   The Settings page appears.

2. Select ZenPacks in the left panel.

   The list of loaded ZenPacks appears.



Pack	Package	Author	Version	Egg
ZenPacks.zenoss.	zenoss	Zenoss	2.0.1	Yes
ZenPacks.zenoss.	zenoss	Zenoss	1.0.0	Yes
ZenPacks.zenoss.	zenoss	Zenoss	1.1.3	Yes
ZenPacks.zenoss.ApacheMonitor	zenoss	Zenoss	2.1.2	Yes
ZenPacks.zenoss.BigIpMonitor	zenoss	Zenoss	2.2.2	Yes
ZenPacks.zenoss.BrocadeMonitor	zenoss	Zenoss	2.0.4	Yes
ZenPacks.zenoss.CheckPointMonitor	zenoss	Zenoss	1.0.4	Yes
ZenPacks.zenoss.CiscoMonitor	zenoss	Zenoss	2.6.0	Yes
ZenPacks.zenoss.DellMonitor	zenoss	Zenoss	2.1.0	Yes
ZenPacks.zenoss.Diagram	zenoss	Zenoss	1.1.1	Yes
ZenPacks.zenoss.DigMonitor	zenoss	Zenoss	1.0.2	Yes
ZenPacks.zenoss.DistributedCollector	zenoss	Zenoss	2.2.2	Yes
ZenPacks.zenoss.DnsMonitor	zenoss	Zenoss	2.0.2	Yes
ZenPacks.zenoss.EnterpriseCollector	zenoss	Zenoss	1.0.0	Yes
ZenPacks.zenoss.EnterpriseLinux	zenoss	Zenoss	1.1.0	Yes
ZenPacks.zenoss.EnterpriseReports	zenoss	Zenoss	2.1.2	Yes
ZenPacks.zenoss.EnterpriseSecurity	zenoss	Zenoss	1.0.0	Yes
ZenPacks.zenoss.EnterpriseSkin	zenoss	Zenoss	2.0.1	Yes
ZenPacks.zenoss.FtpMonitor	zenoss	Zenoss	1.0.2	Yes

Figure 1.1. Loaded ZenPacks

From the action menu on this page, you can create, install, and delete ZenPacks.

Note

Alternatively, you can view loaded ZenPacks from the command line:

```
zenpack --list
```

1.3. Creating ZenPacks

Read the following information and procedures to learn more about why you might want to create a ZenPack, and how to:

- Create a ZenPack
- Add a database object to a ZenPack
- View database objects in a ZenPack
- Remove a database object from a ZenPack
- Add other items to a ZenPack

1.3.1. Why Create a ZenPack?

Suppose you have developed a monitoring template for a new piece of hardware. You have created data sources for the OID's you think are worth monitoring, thresholds to make sure some of these values stay within reasonable limits, and several graph definitions to show this data graphically. Perhaps you also have created a new device class for this hardware. You can create a ZenPack to easily distribute your template and device class to other administrators. This ZenPack can be entirely created from within the user interface.

As another example, suppose you want to monitor a new piece of software running on one of your servers. You would like to monitor several performance metrics of this software, but they are available only via a programmatic API provided with the software. You could develop a new collector daemon to gather data via this API and provide it back to the system. You might also create a new type of data source to provide configuration data for the new collector. Obviously this effort would require development skills and intimate knowledge of the system not necessary for the previous example, but this functionality can be distributed as a ZenPack.

1.3.2. Create a ZenPack

Use the following instructions and guidelines to create a ZenPack.

Note

You must be logged in as an administrator to create a ZenPack.

1. From the navigation bar, select Advanced > Settings.
2. In the left panel, select ZenPacks.
3. From ⚙▾ (Action menu), select Create a ZenPack.

   The Create a ZenPack dialog appears.
4. Enter the name of the ZenPack, which must be in the format:

   ZenPacks.*Organization*.*Identifier*

   where *Organization* is a name that identifies you or your organization and *Identifier* is a string that represents the intent of your ZenPack.
5. Click **OK**.

   The system creates the ZenPack object in the database and a new directory in the file system `$ZENHOME/ZenPacks/`*YourZenPackID*.

1.3.3. Add a Database Object to a ZenPack

To add a database object (such as a device, service, or event class, event mapping, user or event command, device organizer, or monitoring template) to a ZenPack:

1. Navigate to the object in the interface.
2. From the Action menu, select Add to ZenPack.

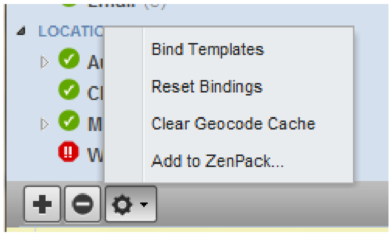


Figure 1.2. Add to ZenPack

The Add to ZenPack dialog appears.

3. Select a ZenPack from the list of installed ZenPacks, and then click **Submit**.

1.3.4. View Database Objects in a ZenPack

To view the objects that are part of a ZenPack:

1. From the navigation bar, select Advanced > Settings.
2. In the left panel, select ZenPacks.
3. Click the name of a ZenPack in the list.

   The ZenPack Provides area of the page lists objects that are part of the ZenPack.

1.3.5. Remove a Database Object from a ZenPack

To remove a database object from a ZenPack:

1. From the navigation bar, select Advanced > Settings.
2. In the left panel, select ZenPacks.
3. Click the name of a ZenPack in the list.
4. Select an object in the ZenPack Provides area of the page.
5. From the Action menu, select Delete from ZenPack.

1.3.6. Adding Other Items to ZenPacks

ZenPacks can contain items that are not ZEO database items, such as:

- Daemons
- Data source types
- Skins

You can add these to a ZenPack by placing them in the appropriate subdirectory in the ZenPack's directory. See the Core ZenPacks at http://community.zenoss.org/community/zenpacks for examples of how to incorporate such items into your ZenPack.

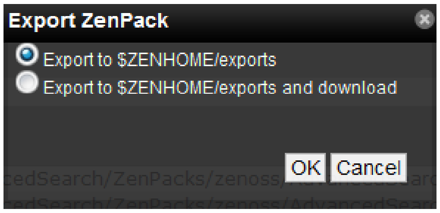
1.4. Packaging and Distributing ZenPacks

Follow these steps to create an installable ZenPack .egg file:

1. From the navigation bar, select Advanced > Settings.
2. In the left panel, select ZenPacks.

3. Click the name of a ZenPack in the list.
4. From ⚙ (Action menu) located at the bottom left of the page, select Export ZenPack.

   The Export ZenPack dialog appears.

   Export ZenPack

   ● Export to $ZENHOME/exports

   ○ Export to $ZENHOME/exports and download

   Figure 1.3. Export ZenPack
5. Select one of the export options:
   - **Export to $ZENHOME/exports** - Exports the ZenPack to a file named *ZenPackID*`.egg` in the `$ZENHOME/exports` directory on the Zenoss server.
   - **Export to $ZENHOME/exports** - Additionally downloads the exported file to your browser. Other administrators can then install this exported `.egg` file.
6. Click **OK**.

1.5. Removing ZenPacks

Warning

Removing a ZenPack can have unexpected consequences. For example, removing a ZenPack that installs a device class removes the device class and all devices in that class.

Before removing a ZenPack, you should:

- Delete any data source of a type provided by the ZenPack
- Perform a backup of your system data. (See the section titled "Backup and Restore" in *Zenoss Administration* for information on backing up your system data.)

1.6. Where to Find More Information

Further information about ZenPack development is available in the *Zenoss Developer's Guide*.

Discussions about ZenPack development and implementation take place on the zenoss-dev and zenoss-zenpacks forums:

- http://community.zenoss.org/community/forums
- http://community.zenoss.org/community/developers/zenpack_development
- http://zenpacks.zenoss.org

Part I. Core ZenPacks

Chapter 2. Amazon Web Services

2.1. About

The Amazon Web Services™ (AWS™) ZenPack allows you to monitor Amazon Elastic Compute Cloud™ (Amazon EC2™) server instances. It collects information for these objects monitored through Amazon's CloudWatch APIs:

- Account
- Instance
- Instance Type

When you install the ZenPack, the `/AWS/EC2` device class is added to your Zenoss instance. A single device in the EC2 class, `EC2Manager`, represents your EC2 account. All instances and instance types are contained in the EC2 account manager.

2.2. Prerequisites

You must have a valid Amazon Web Services account with the Elastic Compute Cloud service enabled.

Modeling and performance requests to Amazon are sent via XML over http or https. You must open port 80, port 443, or both on your Zenoss server so that requests can be sent to Amazon's infrastructure through the Internet.

The Zenoss server time must be correct; otherwise, no performance data will be returned.

Prerequisite	**Restriction**
Zenoss Version	Zenoss Version 2.5 or higher
Required ZenPacks	AWS

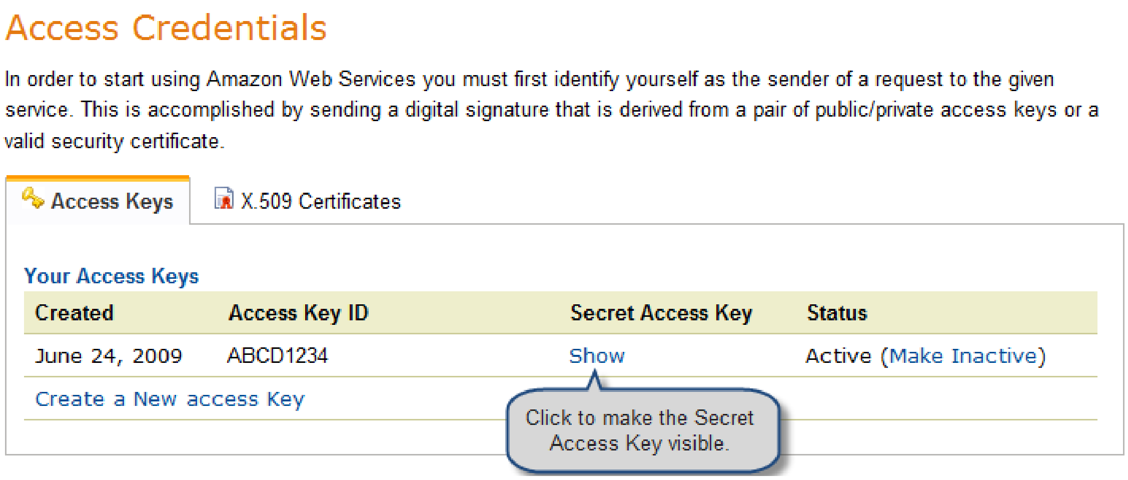
Table 2.1. Prerequisites

2.3. Setup

To set up the EC2 account manager in Zenoss, follow these steps:

1. Retrieve your Amazon EC2 access credentials.
   a. Browse to http://aws.amazon.com.
   b. Select **Security Credentials** from the **Your Account** list of options.

      The Access Key ID and Secret Access Key values appear on the Access Keys tab.



Access Credentials

In order to start using Amazon Web Services you must first identify yourself as the sender of a request to the given service. This is accomplished by sending a digital signature that is derived from a pair of public/private access keys or a valid security certificate.

Your Access Keys

Created	Access Key ID	Secret Access Key	Status
June 24, 2009	ABCD1234	Show	Active (Make Inactive)

Create a New access Key

Click to make the Secret Access Key visible.

Figure 2.1. Access Credentials

2. In the Zenoss interface, select Infrastructure, and then select the EC2Manager device in the device list.
3. Select Configure EC2 in the left panel.
4. Enter access credentials.

   Note

   Entering a class for the Device Mapping field allows the system to monitor an EC2 instance as a normal device. If no class is specified, then the instances are monitoring within EC2Manager only.

5. Model the EC2Manager account to retrieve the Instance and InstanceType objects.

2.4. Working with the EC2Manager Account

Select Infrastructure, and then select the EC2Manager account in the device list. Select Instances in the left panel to see each instance that is active in your Amazon EC2 account. Click an instance to view detailed information

The Instance Type field is a link to a type object that references all instances of a particular type.

2.4.1. CloudWatch Data

Amazon provides monitoring information through its CloudWatch APIs. These APIs provide monitoring information for each of their primary objects (Account, Instance, and Instance Type).

Metrics provided by the API are:

- CPU utilization
- Network in/out
- Disk bytes read/write
- Disk operations read/write

The metrics for an instance apply directly for the instances; for example, if an instance shows 100% CPU utilization, then its CPU is at maximum. However, for an instance type, 100% CPU utilization means that all instances within that type are at 100% CPU utilization. The same is true for the account; metrics are summed for all instances.

Zenoss collects monitoring information for the Account, Instance, and Instance Type objects. Account information appears on the Perf tab. Instance and Instance Type information appears on their main screens.

2.4.2. Templates and Collection

Zenoss uses the standard monitoring template system to configure EC2 Manager accounts. Each template relies on a custom ZenCommand, `zencw2`, that polls the CloudWatch API and returns all available parameters. These parameters are used by their associated graphs. You can set thresholds against the parameters.

Templates for each primary object type are defined in the `/AWS/EC2` class.

Object Type	Template
Account	EC2Manager
Instance	EC2Instance
Instance Type	EC2InstanceType

Table 2.2. Primary Object Type Templates

2.4.2.1. Example: Initiating Load-Based Elasticity for an EC2 Setup

Suppose you want to use Zenoss to initiate load-based elasticity for your EC2 setup. For example, each time the account CPU exceeds 80%, you want Zenoss to create a new instance. To set up this scenario, you would first set up and model your account. Then, you would follow these steps:

1. Select the EC2Manager device in the Devices section of the Infrastructure page, and then expand the Monitoring templates node at the left of screen and click the EC2Manager template.
2. Add a threshold against the `zencw2_CPUUtilization` CPU Utilization data point, and then set its event class to `/Perf/CPU`.

   Each time the CPU exceeds the threshold, Zenoss creates an event with the device name EC2Manager in the `/Perf/CPU` class.
3. Create an event command that matches this event, and launch the EC2 command to create the new instances.

   When the event is initiated, the new instances will be created.

   Note

   The `clear` command can be used to shut down unneeded instances.

Chapter 3. Apache Web Server

3.1. About

The ApacheMonitor ZenPack provides a method for pulling performance metrics from the Apache Web server directly into Zenoss, without requiring the use of an agent. This is accomplished by using Apache's `mod_status` module that comes with Apache Version 1 and 2.

The following metrics are collected and graphed for the 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)

3.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.ApacheMonitor

Table 3.1. Apache Prerequisites

3.3. Enable Monitoring

Follow the steps in these sections to:

- Display the status page in Apache Version 1.3 or higher
- Display the status page in Apache Version 2.x
- Configure your configuration
- Configure the system to monitor the Web server

3.3.1. 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 Zenoss 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
   ```

3.3.2. 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 Zenoss 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
```

3.3.3. 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 Zenoss server or servers must be able to connect to your Apache server by using HTTP to receive information. This means that the Zenoss server must be permitted not only by the Apache configuration settings, but also by any firewalls or proxies between the Zenoss server and the Apache server, including any firewall on the Apache server. If there are any proxies, they must be configured to allow the Zenoss HTTP traffic through to Zenoss. 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).

3.3.4. Configure Zenoss to Monitor the Web Server

Once the Apache server is configured to allow Zenoss to access the extended status, you can add Apache monitoring to the device within Zenoss 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.

3.4. Daemons

Type	Name
Performance Collector	**zencommand**

Table 3.2. Daemons

Chapter 4. Dell Hardware

4.1. About

The DellMonitor ZenPack provides custom modeling of devices running the Dell OpenManage agents. It also contains hardware identification for Dell proprietary hardware. The information is collected through the SNMP interface.

The following information is modeled:

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

4.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 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.

Table 4.1. Dell Hardware Prerequisites

4.3. 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.

4.4. Daemons

Type	Name
Modeler	**zenmodeler**
Performance Collector	**zenperfsnmp**

Table 4.2. Daemons

Chapter 5. Domain Name System (DNS)

5.1. About

DigMonitor monitors the response time of DNS lookups for devices running a DNS server.

5.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.DigMonitor

Table 5.1. DNS (DigMonitor) Prerequisites

5.3. 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.

Option	Description
DNS Server	Name server against which the **dig** command should be run.
Port	Port on which the name server is listening. This is normally port 53.
Record Name	Name of the record you want to look up.
Record Type	DNS record type (for example, A, MX, CNAME).

Table 5.2. DigMonitor Data Source Options

5.4. Daemons

Type	Name
Performance Collector	**zencommand**

Table 5.3. Daemons

Chapter 6. File Transfer Protocol (FTP)

6.1. About

The FTPMonitor ZenPack monitors connection response time to an FTP server.

6.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.FTPMonitor

Table 6.1. FTP Prerequisites

6.3. 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.

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

Table 6.2. FTPMonitor Basic Data Source Options

6.4. 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.

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

Table 6.3. FTPMonitor Secure Data Source Options

6.5. 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.

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

Table 6.4. FTPMonitor Tunables Data Source Options

6.6. Daemons

Type	Name
Performance Collector	**zencommand**

Table 6.5. Daemons

Chapter 7. HP PC Hardware

7.1. About

HPMonitor provides custom modeling of devices running the HP Insight Management Agents. It also contains hardware identification for HP proprietary hardware. The information is collected through the SNMP interface.

The following information is modeled:

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

7.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.HPMonitor
On each remote device	The HP Insight SNMP Management Agent gathers information about the device.

Table 7.1. HP PC Hardware Prerequisites

7.3. 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.

7.4. Daemons

Type	Name
Modeler	**zenmodeler**
Performance Collector	**zenperfsnmp**

Table 7.2. Daemons

Chapter 8. Internet Relay Chat (IRC)

8.1. About

ZenPacks.zenoss.IrcdMonitor monitors the number of users connected to an IRC server.

8.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.IrcdMonitor

Table 8.1. IRC Prerequisites

8.3. 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 IrcdeMonitor 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.

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

   Table 8.2. IRC Basic Data Source Options

8.4. Daemons

Type	Name
Performance Collector	**zencommand**

Table 8.3. Daemons

Chapter 9. Jabber Instant Messaging

9.1. About

ZenPacks.zenoss.JabberMonitor monitors the response time of devices running a Jabber server.

9.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.JabberMonitor

Table 9.1. Jabber Prerequisites

9.3. 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.

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.
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>`

Table 9.2. Jabber Data Source Options

9.4. Daemons

Type	Name
Performance Collector	**zencommand**

Table 9.3. Daemons

Chapter 10. Java 2 Platform Standard Edition (J2E)

10.1. About

ZenJMX is a ZenPack that allows Zenoss to communicate with remote Java Management Extensions (JMX) agents. The ZenJMX 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

ZenJMX also includes a built-in template named `ZenJMX`. This template should only be used on the device running the **zenjmx** daemon. To monitor other Java servers use the included `Java` template.

When the **zenjmx** daemon is started it communicates with ZenHub and retrieves a list of devices that possess `JMX` data sources. It also spawns a Java process. ZenJMX 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 and returns the results to ZenJMX. Any collection or configuration errors are sent as events to Zenoss and will appear in the event console.

Lastly, ZenJMX heartbeats after each collect to ZenHub to let Zenoss know that ZenJMX is still alive and well.

10.1.1. 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** (Sun Microsystems' 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, Sun introduced the Remote API for Java Management Extensions. This enhancement defines an RMI wrapper around a JMX agent and allows for independent client development. ZenJMX 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.

10.1.2. ZenJMX Capabilities

ZenJMX 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 ZenJMX 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 Zenoss and graphed and checked against thresholds. Non-numerical values (Strings and complex types) cannot be retrieved and stored by Zenoss.

When setting up data points, make sure you understand the semantics of the attribute name and choose the correct Zenoss 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).

10.1.3. 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`

10.1.4. 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 Section 10.5, “Using **JConsole** to Query a JMX Agent” for additional information on how to inspect the MBeans deployed in an JMX Agent.

10.1.5. 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 Zenoss with the following information:

- MBean Name: java.lang:type=Memory
- Attribute Name: HeapMemoryUsage
- Data Points:
    - committed (type: gauge)
    - used (type: gauge)
    - max (type: gauge)

10.1.6. 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()".

ZenJMX has the ability to invoke operations, but there are some subtleties in how ZenJMX sends parameters to the JMX Agent and interprets the response.

10.1.6.1. 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)

10.1.6.2. 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 Zenoss. 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.

ZenJMX 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 ZenJMX follows:

- MBean Name: Application:Name=Accounting,Type=Accounting
- Operation Name: getBalanceSummary()
- Data Points:
  - dailyBalance (type: gauge)
  - annualBalance (type: gauge)

10.1.6.3. 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)

10.1.6.4. 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[].

ZenJMX 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)

10.1.6.5. Multiple parameters in polymorphic operations

The above 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.

10.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Zenoss Product	Zenoss Core and Zenoss Enterprise
Required ZenPacks	ZenPacks.zenoss.ZenJMX
Other	Sun JRE Version 5.0 or higher

Table 10.1. J2EE Prerequisites

10.2.1. Sun Java Runtime Environment (JRE)

ZenJMX requires Sun JRE Version 5.0 or higher. Make sure that after you install Sun's 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)
```

Warning

Sun's Java version 5 (aka 1.5) **must** be installed. The GNU Java does not work.

Note

ZenJMX and Sun's JRE is installed using a **conary** command. As root, run the following command:

```
conary update --resolve group-zenjmx
```

10.3. Example to Monitor a JMX Value

10.3.1. Enabling Remote JMX Access

Each application server has a slightly different process for enabling remote JMX Access. It's best to consult with your application server for specific instructions. We've included instructions for a few commonly used configurations below.

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 the Remote Method Invocation (RMI) protocol 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 (via RMI). ZenJMX can only be used with remote servers via RMI and cannot work with local-only serialized MBeans. This is not a significant limitation because ZenJMX can establish RMI connections to localhost just as easily as it can establish RMI 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.

10.3.2. Configure Zenoss with a Custom Data Source

Custom JMX data sources allow system administrators to monitor any attribute or operation result accessible via a JMX call. ZenJMX 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 ZenJMX.

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**.

Option	Description
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`

Table 10.2. Memory Head Example ZenJMX Data Source Options

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. (For more information, see *Zenoss Administration*.)

Review Section 10.5, “Using **JConsole** to Query a JMX Agent” to learn how to determine the object name, attribute name, and data points that might be interesting in your application.

10.4. 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.

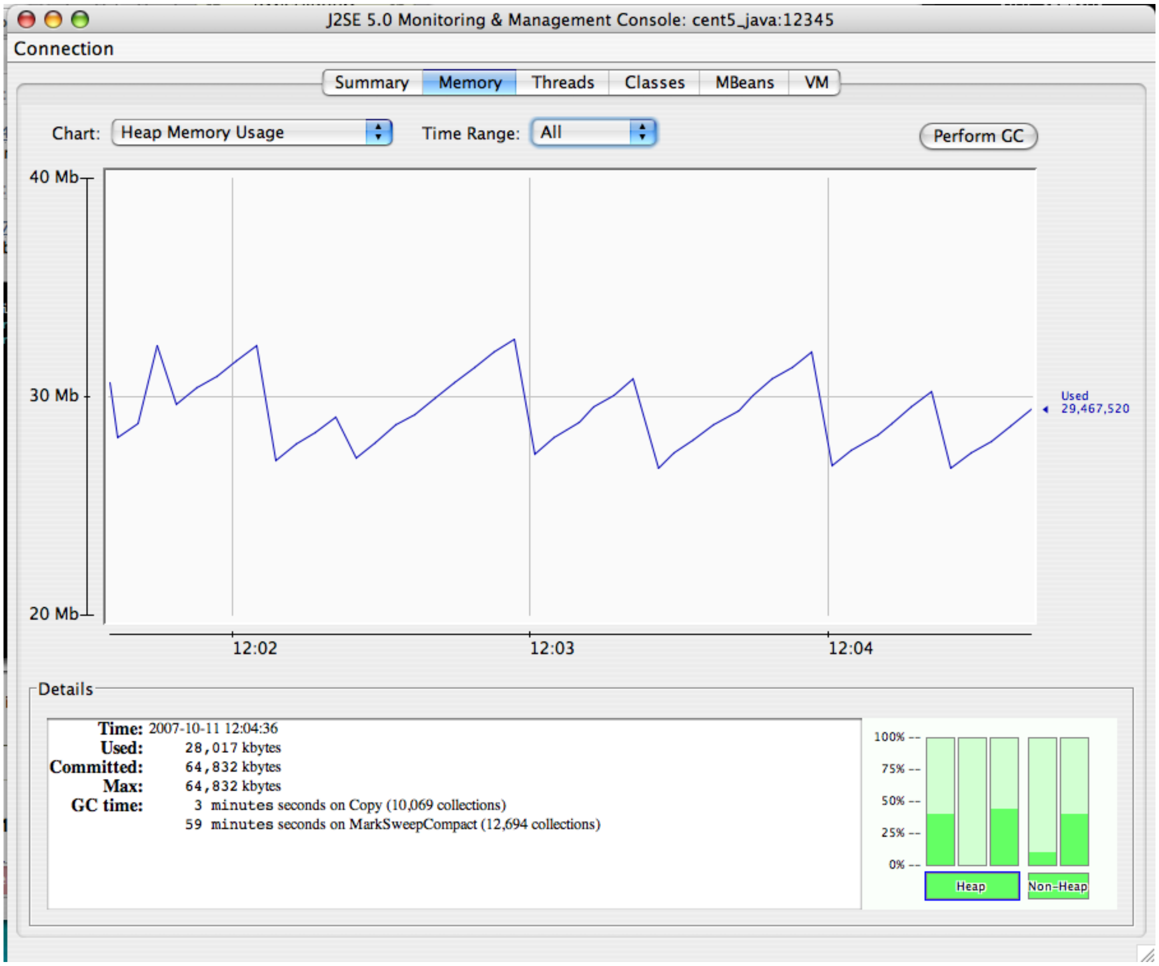
In this example, since the index being used for the tabular data is not a multi-value index and so the column name is optional. The Attribute Path can be written as:

```
memoryUsageAfterGc.[Tenured Gen].used
```

10.5. 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.



J2SE 5.0 Monitoring & Management Console: cent5_java:12345

Chart: Heap Memory Usage

Time Range: All

Details

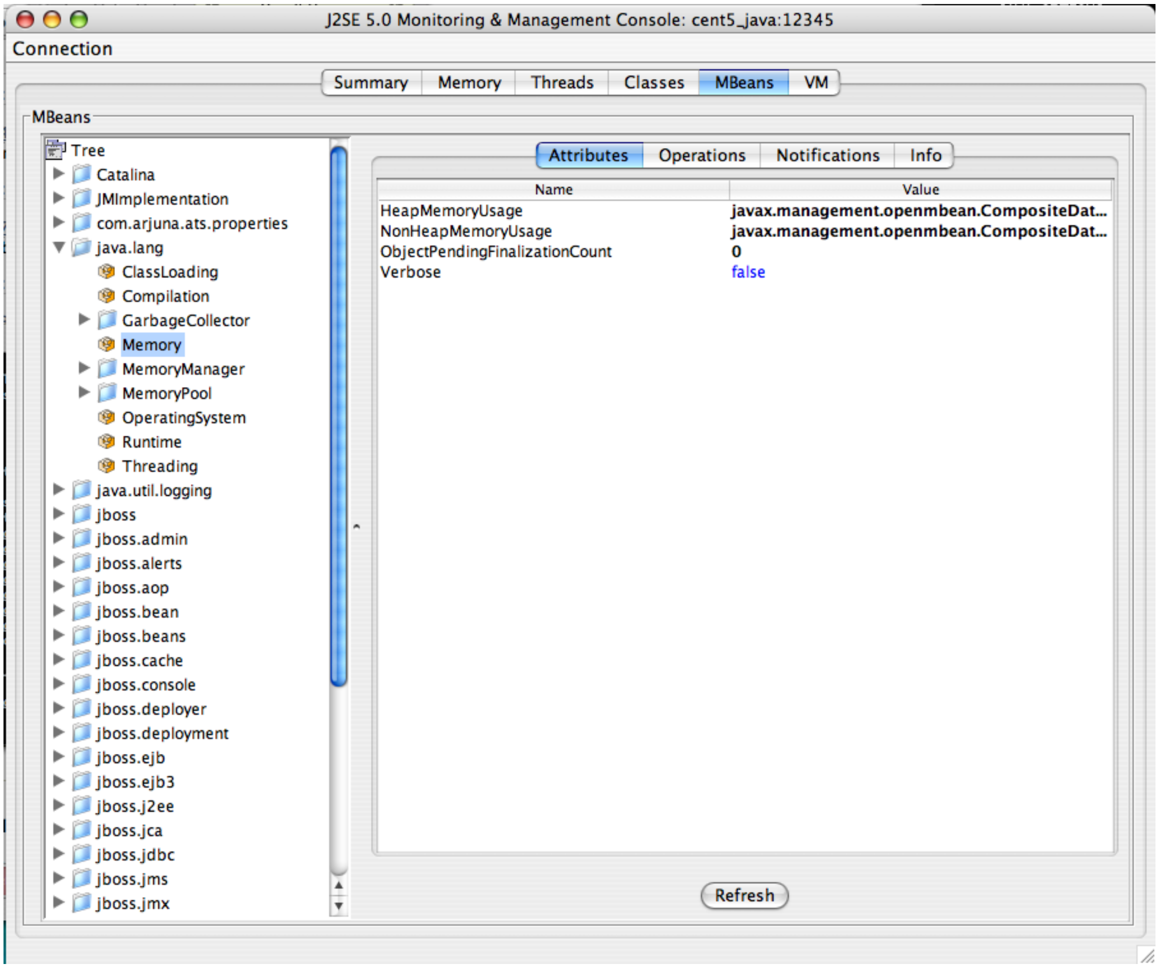
Time: 2007-10-11 12:04:36
Used: 28,017 kbytes
Committed: 64,832 kbytes
Max: 64,832 kbytes
GC time: 3 minutes seconds on Copy (10,069 collections)
59 minutes seconds on MarkSweepCompact (12,694 collections)

Figure 10.1. 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).

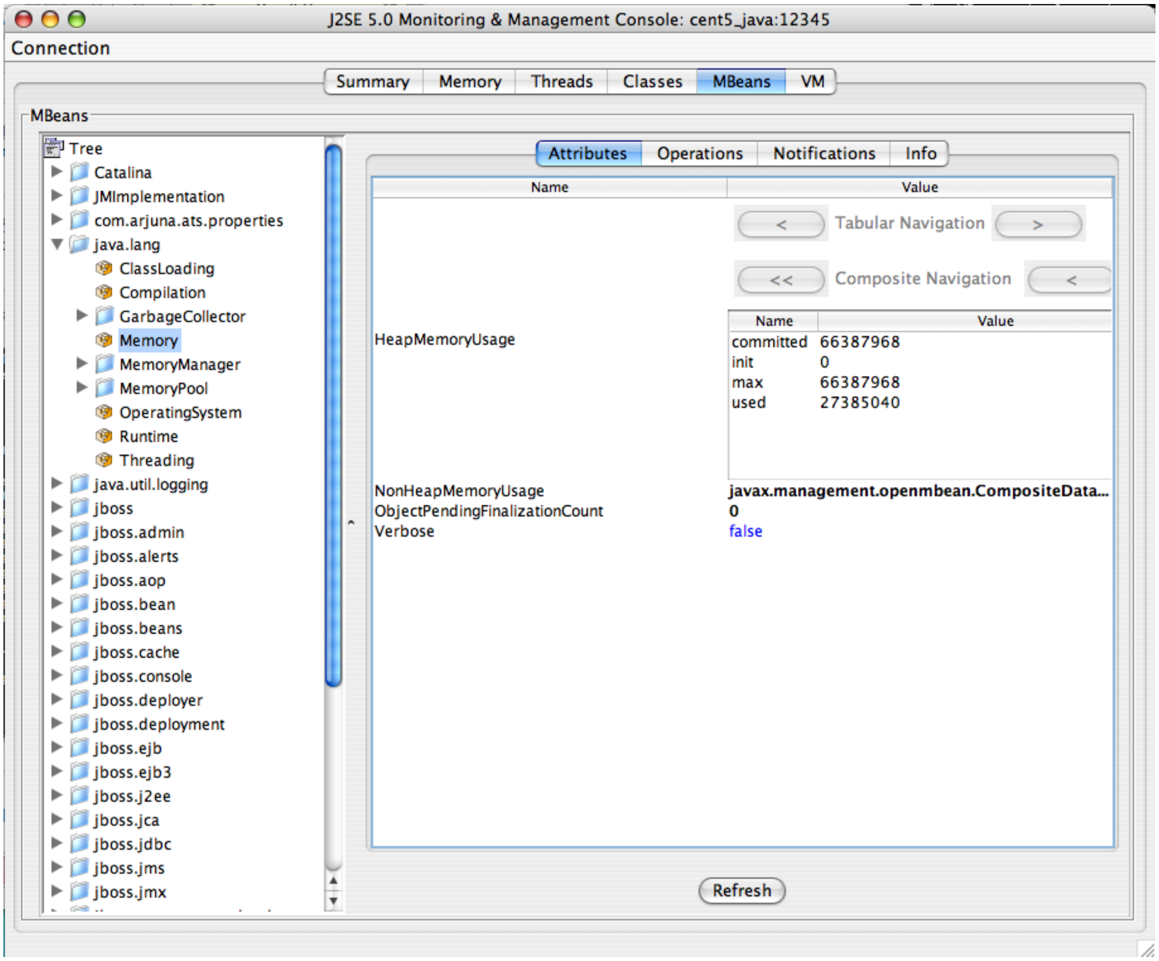
To get started expand the java.lang node in the Tree. This will expose several MBeans as well as additional folders. Click on the Memory MBean and observe how the right hand side of the panel is populated with information about the Memory MBean.



Name	Value
HeapMemoryUsage	**javax.management.openmbean.CompositeDat...**
NonHeapMemoryUsage	**javax.management.openmbean.CompositeDat...**
ObjectPendingFinalizationCount	**0**
Verbose	false

Figure 10.2. 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 Zenoss. 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.



Name	Value
HeapMemoryUsage	
NonHeapMemoryUsage	javax.management.openmbean.CompositeData...
ObjectPendingFinalizationCount	0
Verbose	false

Name	Value
committed	66387968
init	0
max	66387968
used	27385040

Figure 10.3. 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.

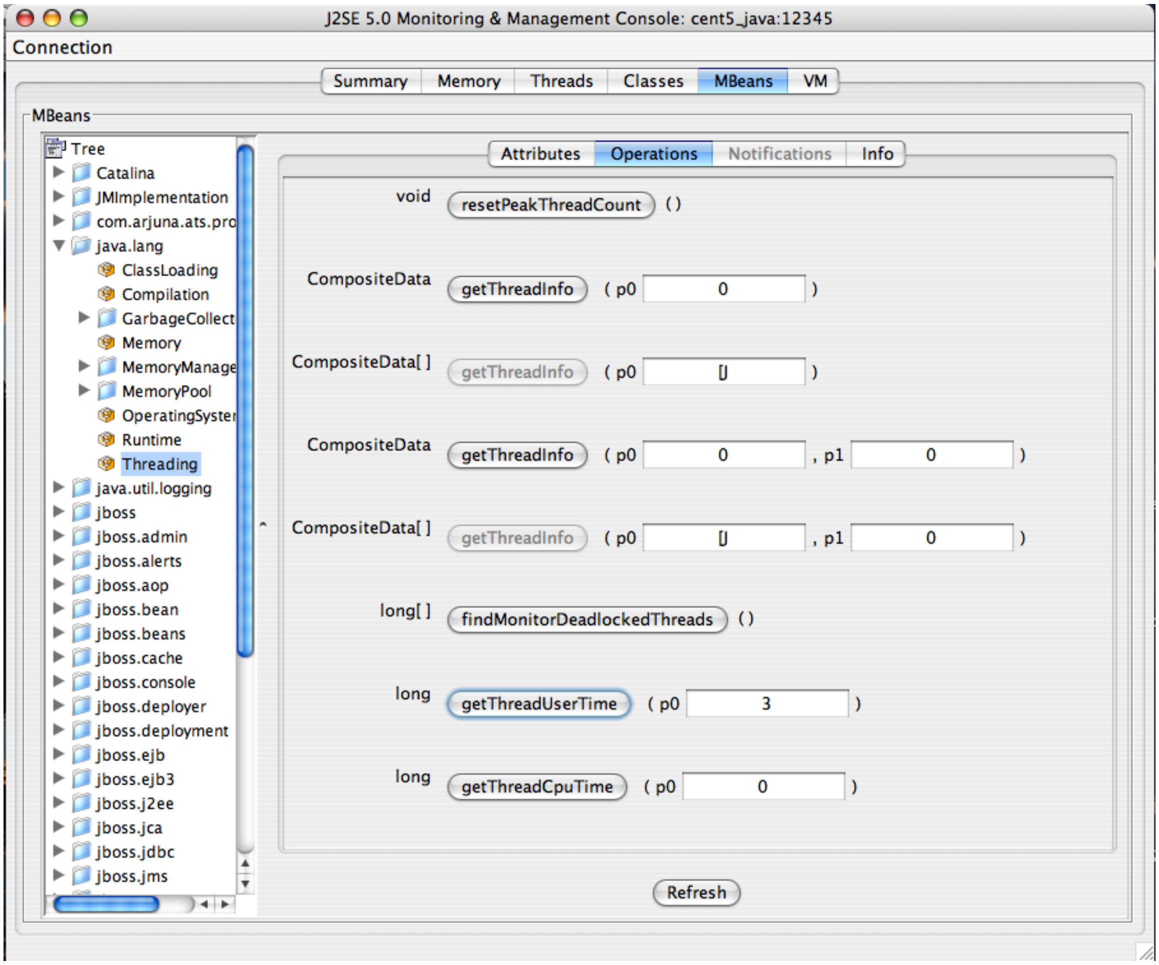


Figure 10.4. Operations Tab

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.

10.6. Daemons

Type	Name
Performance Collector	**zenjmx**

Table 10.3. Daemons

Chapter 11. Lightweight Directory Access Protocol (LDAP) Response Time

11.1. About

ZenPacks.zenoss.LDAPMonitor monitors the response time of an LDAP server (in milliseconds).

11.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.LDAPMonitor

Table 11.1. LDAP Monitoring Prerequisites

11.3. Enable Monitoring

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

11.3.1. For a Device

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. Select Configuration Properties from the left panel.
4. Modify configuration property values as needed for your environment. Check with your LDAP administration for more information.

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

   Table 11.2. LDAPServer Configuration Properties
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.

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

   Table 11.3. LDAPServer Basic Data Source Options

   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 and observing that the check_ldap or check_ldaps command correctly connects to your LDAP server:

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

11.4. Daemons

Type	Name
Performance Collector	**zencommand**

Table 11.4. Daemons

Chapter 12. MySQL Database

12.1. About

MySqlMonitor provides a method for pulling performance metrics from the MySQL database server directly into Zenoss without requiring the use of an agent. This is accomplished by using the MySQL client library to connect to the database remotely.

The following metrics are collected and graphed for MySQL server:

- Command Statistics (SELECT, INSERT, UPDATE, DELETE)
- Select Statistics (Scan, Range Check, Range Join, Full Join)
- Handler Statistics (Keyed and Unkeyed Reads, Writes, Updates, Deletes)
- Network Traffic (Received and Sent)

12.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.MySqlMonitor

Table 12.1. MySQL Prerequisites

12.3. Enable Monitoring

Use the following procedures to enable monitoring.

12.3.1. Authorize MySQL Performance Data Access

Follow these steps to set up your MySQL server to allow Zenoss to read performance data from the system tables.

1. Connect to the MySQL database by using the MySQL client:

   ```
   mysql -u root
   ```

   Alternatively, if there is a MySQL root password:

   ```
   mysql -u root -p
   ```

2. Create a user for Zenoss to use. The username "zenoss" is recommended.

   ```
   mysql> CREATE USER zenoss IDENTIFIED BY 'zenossPassword';

   Query OK, 0 rows affected (0.00 sec)
   ```

12.3.2. Set up Zenoss

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. Edit the zMySqlRootPassword configuration property for the device or devices in Zenoss on which you want to monitor MySQL.
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 MySQL template to the list of selected templates, and then click **Submit**.

   The MySQL template is added to the list of monitoring templates.

Note

Pay particular attention to the MySQL Version 5+ setting in the data source. If you are monitoring pre-v5 installations of MySQL, then you must set this value to False. If you are monitoring pre-v5 and v5+ installations, then create two templates: one for MySQL installations earlier than v5 and another for those after.

12.4. Daemons

Type	Name
Performance Collector	**zencommand**

Table 12.2. Daemons

Chapter 13. Network News Transport Protocol (NNTP)

13.1. About

ZenPacks.zenoss.NNTPMonitor ZenPack monitors the response time of an NNTP server in milliseconds.

13.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.NNTPMonitor

Table 13.1. NNTP Prerequisites

13.3. 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
   ```

13.4. Daemons

Type	Name
Performance Collector	**zencommand**

Table 13.2. Daemons

Chapter 14. Network Time Protocol (NTP)

14.1. About

ZenPacks.zenoss.NtpMonitor monitors the offset between system time and a target NTP (Network Time Server) server's time.

14.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.NtpMonitor

Table 14.1. NTP Prerequisites

14.3. 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.

14.4. Daemons

Type	Name
Performance Collector	**zencommand**

Table 14.2. Daemons

Chapter 15. ONC-style Remote Procedure Call (RPC)

15.1. About

ZenPacks.zenoss.RPCMonitor monitors the availability of an ONC RPC server.

15.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.RPCMonitor

Table 15.1. ONC RPC Prerequisites

15.3. 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.

15.4. Daemons

Type	Name
Performance Collector	**zencommand**

Table 15.2. Daemons

Chapter 16. SSH Monitoring Example

16.1. About

The LinuxMonitor ZenPack demonstrates the new Secure Shell (SSH) features. This example ZenPack includes functionality to model and monitor 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 Zenoss server or on a distributed collector. The account used to monitor the device does not require root access or special privileges.

This ZenPack is provided for developers as it provides some examples of how to create SSH performance collecting plugins. See the *Zenoss Developer's Guide* for more information about the new SSH features.

16.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.4 or higher
Required ZenPacks	ZenPacks.zenoss.LinuxMonitor

Table 16.1. Linux SSH Monitoring Example Prerequisites

16.3. Set Linux Server Monitoring Credentials

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

Tip

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.

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

Table 16.2. Linux Configuration Properties

16.4. 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:

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 Section 16.3, “Set Linux Server Monitoring Credentials”) and then manually model the device.

Table 16.3. Adding Linux Device Details

4. Click **Add**.

16.5. Daemons

Type	Name
Modeler	**zenmodeler**
Performance Collector	**zencommand**

Table 16.4. Daemons

Chapter 17. VMware Infrastructure ESX Server

17.1. About

The EsxTop ZenPack uses the `resxtop` command to gather performance information about VMware Infrastructure™ ESX™ servers. A basic modeler creates virtual machines under the `/Devices/Server/Virtual Hosts/EsxTop` device class for any host device that is added and modeled.

17.2. Prerequisites

To implement this ZenPack, you must:

- Install the OpenSSL development package, Version 0.9.7 or higher
- Install the VMware vSphere CLI (as described in the section titled Installing Prerequisite Libraries).
- Update the ZenossVirtualHostMonitor ZenPack to Version 2.3.5.

Prerequisite	Restriction
Zenoss Version	Zenoss Version 3.0 or higher
Required ZenPacks	ZenPacks.zenoss.ZenossVirtualHostMonitor ZenPacks.zenoss.EsxTop
VMware vSphere™ Command-Line Interface (CLI)	VMware vSphere CLI Version 4.1 or higher must be installed on the Zenoss collectors.

Table 17.1. Prerequisites

17.2.1. Installing Prerequisite Libraries

The VMware vSphere CLI is required for access to the `resxtop` command, which enables Zenoss 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 Zenoss 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
```

17.3. Enabling the ZenPack

Follow these steps to set up the EsxTop ZenPack. From the Zenoss interface, add a host:

1. From Infrastructure > Devices, navigate to the `/Devices/Server/Virtual Hosts/EsxTop` device class.
2. From 🖥, 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 ⚙ (Action menu), select Model device.

17.4. Daemons

Type	**Name**
Modeler	**zenmodeler**
Performance Collector	**zencommand**

Table 17.2. Daemons

Chapter 18. Web Page Response Time (HTTP)

18.1. About

ZenPacks.zenoss.HttpMonitor monitors connection response time to an HTTP server and determines whether specific content exists on a Web page.

18.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.HttpMonitor

Table 18.1. HTTP Prerequisites

18.3. 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 Zenoss release is prior to Zenoss 2.4 you must create the template, data source and graphs manually. See *Zenoss Administration* for more details on these steps.
6. The HttpMonitor template is added to the list of monitoring templates. You can now begin collecting Web server metrics from the device.

18.4. 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**.

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`.

Table 18.2. HTTPMonitor Content Checking Data Source Options

18.5. Check for Specific Content on the Web Page

This procedure allows Zenoss 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**.

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?
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.

Table 18.3. HTTPMonitor Content Checking Data Source Options

18.6. 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**.

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)

Table 18.4. HTTPMonitor Tunables Data Source Options

18.7. Daemons

Type	Name
Performance Collector	**zencommand**

Table 18.5. Daemons

Chapter 19. Xen Virtual Hosts

19.1. About

The XenMonitor ZenPack allows you to monitor Xen para-virtualized domains with Zenoss.

This ZenPack:

- Extends ZenModeler to discover guests running on the Xen host.
- Provides screens and templates for collecting and displaying resources allocated to guests.

The XenMonitor ZenPack requires the ZenossVirtualHostMonitor ZenPack to be installed as a prerequisite.

19.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.XenMonitor ZenPacks.zenoss.ZenossVirtualHostMonitor

Table 19.1. Xen Virtual Hosts Prerequisites

19.3. 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 Zenoss server to log in as root without requiring further credentials.
2. Create the Xen server in the `/Servers/Virtual Hosts/Xen` device class.

   Warning

   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.

19.4. Daemons

Type	Name
Modeler	**zenmodeler**
Performance Collector	**zencommand**

Table 19.2. Daemons

Part II. Enterprise ZenPacks

Chapter 20. Advanced Search

20.1. About

The AdvancedSearch ZenPack enables the advanced search facility in the user interface. This tool allows you to locate devices and other system objects, as well as events and services.

When enabled, advanced search appears adjacent to the user information area.

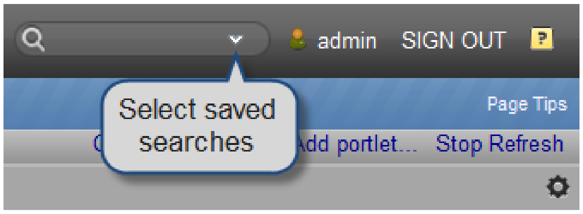
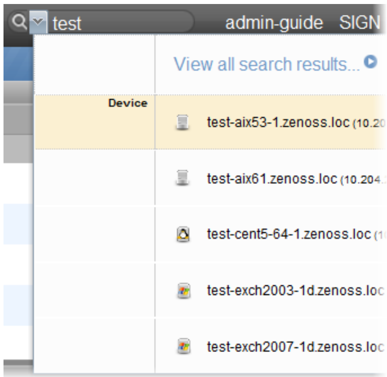


Figure 20.1. Advanced Search (User Information Area)

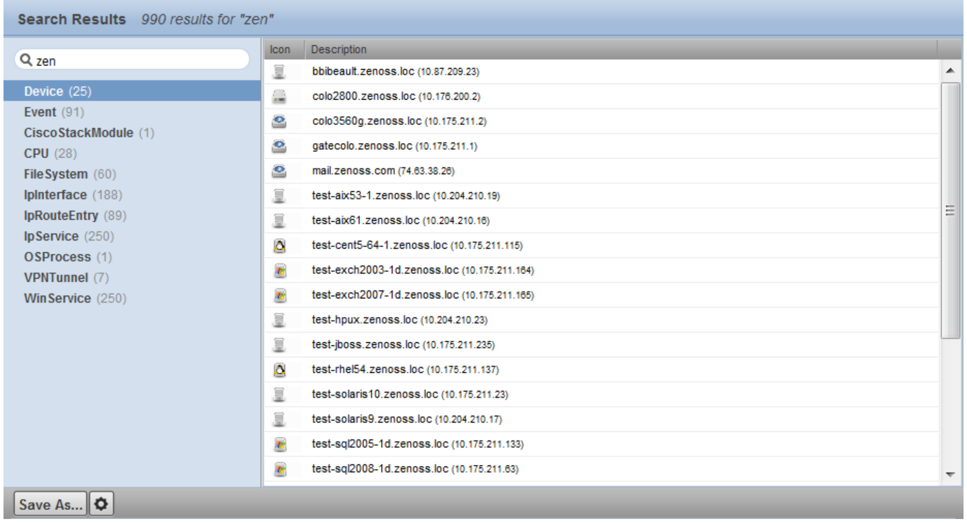
To search, enter part or all of a name in the search box. The system displays matches, categorized by type.



Device
test-aix53-1.zenoss.loc (10.20
test-aix61.zenoss.loc (10.204
test-cent5-64-1.zenoss.loc (1
test-exch2003-1d.zenoss.loc
test-exch2007-1d.zenoss.loc

Figure 20.2. Search Results

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


Search Results *990 results for "zen"*

zen

- Device (25)
- Event (91)
- CiscoStackModule (1)
- CPU (28)
- FileSystem (60)
- IpInterface (188)
- IpRouteEntry (89)
- IpService (250)
- OSProcess (1)
- VPNTunnel (7)
- WinService (250)

Icon	Description
	bbibeault.zenoss.loc (10.87.209.23)
	colo2800.zenoss.loc (10.176.200.2)
	colo3560g.zenoss.loc (10.175.211.2)
	gatecolo.zenoss.loc (10.175.211.1)
	mail.zenoss.com (74.63.38.26)
	test-aix53-1.zenoss.loc (10.204.210.19)
	test-aix61.zenoss.loc (10.204.210.16)
	test-cent5-64-1.zenoss.loc (10.175.211.115)
	test-exch2003-1d.zenoss.loc (10.175.211.164)
	test-exch2007-1d.zenoss.loc (10.175.211.165)
	test-hpux.zenoss.loc (10.204.210.23)
	test-jboss.zenoss.loc (10.175.211.235)
	test-rhel54.zenoss.loc (10.175.211.137)
	test-solaris10.zenoss.loc (10.175.211.23)
	test-solaris9.zenoss.loc (10.204.210.17)
	test-sql2005-1d.zenoss.loc (10.175.211.133)
	test-sql2008-1d.zenoss.loc (10.175.211.63)

Figure 20.3. All Search Results

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

20.1.1. 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.

20.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 3.0 or higher
Required ZenPacks	ZenPacks.zenoss.AdvancedSearch

Table 20.1. Prerequisites

Chapter 21. AIX

21.1. About

The AixMonitor ZenPack enables Zenoss to use Secure Shell (SSH) to monitor AIX hosts. Zenoss models and monitors devices placed in the `/Server/SSH/AIX` device class by running commands and parsing the output. Parsing of command output is performed on the Zenoss server or on a distributed collector. The account used to monitor the device does not require root access or special privileges.

Specifically, the AixMonitor 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)

21.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.4 or higher
Required ZenPacks	ZenPacks.zenoss.AixMonitor
AIX Releases Supported	5.3 and 6.1

Table 21.1. AIX Prerequisites

Note

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

21.3. 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:

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 (see Section 21.4, “Set AIX Server Monitoring Credentials”) and then manually model the device.

Table 21.2. Adding AIX Device Information

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

21.4. 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:
     a. Select the class in the devices hierarchy.
     b. Click **Details**.
     c. Select Configuration Properties.
   - If applying changes to a device:
     a. Click the device in the device list.
     b. Select Configuration Properties.
2. Verify the credentials for the service account to access the service.

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

Table 21.3. AIX Configuration Properties

3. Click Save to save your changes.

21.5. 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 Zenoss interface.
   - If applying changes to a device class:
     a. Select the class in the devices hierarchy.
     b. Click **Details**.
     c. Select Configuration Properties.
   - If applying changes to a device:

   a. Click the device in the device list.
   b. Select Configuration Properties.

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

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

Table 21.4. Concurrent SSH Configuration Properties

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

21.6. 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 in order to return results from the commands. In order 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 Zenoss interface.
   - If applying changes to a device class:
     a. Select the class in the devices hierarchy.
     b. Click **Details**.
     c. Select Configuration Properties.
   - If applying changes to a device:
     a. Click the device in the device list.
     b. Select Configuration Properties.
2. Apply an appropriate value for the command timeout.

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

Table 21.5. SSH Timeout Configuration Properties

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

21.7. Daemons

Type	Name
Modeler	**zenmodeler**
Performance Collector	**zencommand**

Table 21.6. Daemons

Chapter 22. Apache Tomcat Application Server

22.1. About

TomcatMonitor is a ZenPack that allows System Administrators to monitor the Tomcat Application Server. 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.

TomcatMonitor 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.

TomcatMonitor 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

Tip

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, the TomcatMonitor ZenPack can be used to monitor Tomcat MBeans that are active within JBoss.

22.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher

Prerequisite	Restriction
Required ZenPacks	ZenPacks.zenoss.ZenJMX, ZenPacks.zenoss.TomcatMonitor

Table 22.1. Tomcat Prerequisites

22.3. Enable Monitoring

22.3.1. 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.

22.3.2. Configuring Zenoss

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 Section 10.2.1, “Sun Java Runtime Environment (JRE)” 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 Zenoss interface.
    - If applying changes to a device class:
        a. Select the class in the devices hierarchy.
        b. Click **Details**.
        c. Select Configuration Properties.
    - If applying changes to a device:
        a. Click the device in the device list.
        b. Select Configuration Properties.
2. Edit the appropriate configuration properties for the device or devices.

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.

Table 22.2. Tomcat Configuration Properties

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.

Tip

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".

22.4. 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 Zenoss 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**.

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.

Name	**Description**
`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.

Table 22.3. Tomcat Templates

5. Click the OK button to save your changes.

22.5. Viewing Raw Data

See Section 10.5, “Using **JConsole** to Query a JMX Agent” for more information about how to investigate raw data returned from the application.

22.6. Daemons

Type	**Name**
Performance Collector	**zenjmx**

Table 22.4. Daemons

Chapter 23. BEA WebLogic Application Server

23.1. About

WebLogicMonitor allows you to monitor a WebLogic Server. WebLogicMonitor 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.

23.1.1. 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

23.1.2. 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

23.1.3. Data Pool (JDBC) metrics

- Leaked, Total, and Active connections

- Number of requests waiting for a connection
- Number of reconnect failures

23.1.4. Queue (JMS) Metrics

- Bytes received, currently active, and pending in the queue
- Number of queue consumers
- Number of current, pending, and receives messages

23.2. Prerequisites

Prerequisite	**Restriction**
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.ZenJMX, ZenPacks.zenoss.WebLogicMonitor
BEA WebLogic Versions	WebLogic 9.0 or higher

Table 23.1. BEA WebLogic Prerequisites

23.3. Enable Monitoring

23.3.1. 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.

23.3.2. Configuring Zenoss

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 Section 10.2.1, “Sun Java Runtime Environment (JRE)” 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:
     a. Select the class in the devices hierarchy.
     b. Click **Details**.
     c. Select Configuration Properties.
   - If applying changes to a device:
     a. Click the device in the device list.
     b. Select Configuration Properties.
2. Edit the appropriate configuration properties for the device or devices.

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

Table 23.2. WebLogic Configuration Properties

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.

Tip

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".

23.4. 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**.

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	

Name	Description
WebLogic Thread Pool	Threadpool metrics measured per Tomcat connector
WebLogic Timer Service	
WebLogic User Lockouts	

Table 23.3. WebLogic Templates

5. Click the OK button to save your changes.

23.5. Viewing Raw Data

See the Section 10.5, “Using **JConsole** to Query a JMX Agent” section for more information about how to investigate raw data returned back from the application.

23.6. Monitor SSL-Proxied WebLogic Servers

If you are monitoring a Web application running on a BEA 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 Zenoss 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
```

23.7. Daemons

Type	Name
Performance Collector	**zenjmx**

Table 23.4. Daemons

Chapter 24. BIG-IP Network Devices

24.1. About

The Zenoss BIG-IP network device monitoring feature monitors load balancer CPU and memory utilization. It also tracks per-instance metrics for each load-balanced virtual server that is configured.

24.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.BigIPMonitor

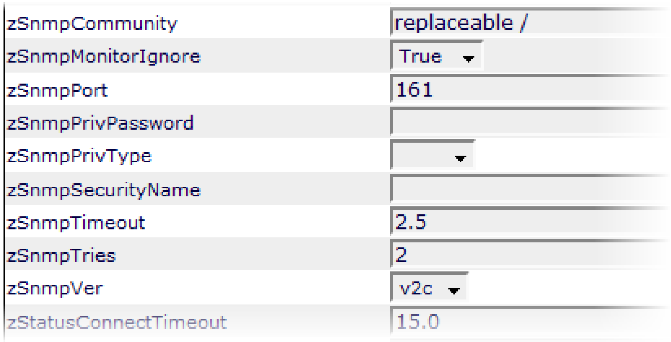
Table 24.1. BIG-IP Prerequisites

24.3. 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`**.

zSnmpCommunity	replaceable /
zSnmpMonitorIgnore	True
zSnmpPort	161
zSnmpPrivPassword	
zSnmpPrivType	
zSnmpSecurityName	
zSnmpTimeout	2.5
zSnmpTries	2
zSnmpVer	v2c
zStatusConnectTimeout	15.0

   Figure 24.1. BIG-IP Configuration Properties Selections
7. Click **Save**.
8. Model the device. To to this, select Manage > Model Device from the page menu.

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

24.4. 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.

24.5. Daemons

Type	Name
Modeler	**zenmodeler**
Performance Collector	**zenperfsnmp**

Table 24.2. Daemons

Chapter 25. Brocade SAN Switches

25.1. About

BrocadeMonitor allows you to monitor Brocade Storage Area Network (SAN) switches.

25.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.BrocadeMonitor, ZenPacks.zenoss.StorageBase

Table 25.1. Brocade Prerequisites

25.3. Enable Monitoring

25.3.1. Configuring Brocade Devices to Allow SNMP Queries

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

25.3.2. Configuring Zenoss

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

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

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`

Table 25.2. Brocade Configuration Properties

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

25.4. Viewing Fibre Channel Port Information

To view the virtual servers, select Brocade Details.

25.5. Daemons

Type	Name
Modeler	**zenmodeler**
Performance Collector	**zenperfsnmp**

Table 25.3. Daemons

Chapter 26. CheckPoint Firewalls

26.1. About

The CheckPointMonitor ZenPack allows you to monitor CheckPoint firewalls.

26.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.CheckPointMonitor

Table 26.1. CheckPoint Prerequisites

26.3. Enable Monitoring

26.3.1. Configuring CheckPoint Firewalls to Allow SNMP Queries

Configure the CheckPoint firewall to allow SNMP queries from the Zenoss server, and to send SNMP v1 or SNMP v2 traps to the Zenoss server.

26.3.2. Configuring Zenoss

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

1. Navigate to the device or device class in the Zenoss interface.
   - If applying changes to a device class:
     a. Select the class in the devices hierarchy.
     b. Click **Details**.
     c. Select Configuration Properties.
   - If applying changes to a device:
     a. Click the device in the device list.
     b. 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.
zSnmpMonitorIgnore	This should be set to `False`
zSnmpPort	The default port is 161
zSnmpVer	This should be set to `v2c`

Table 26.2. CheckPoint Configuration Properties

3. Click Save to save your changes.

   You will now be able to start collecting the CheckPoint 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.

26.4. Daemons

Type	Name
Modeler	**zenmodeler**
Performance Collector	**zenperfsnmp**

Table 26.3. Daemons

Chapter 27. Cisco Devices

27.1. About

The CiscoMonitor ZenPack allows you to monitor a variety of devices from Cisco Systems. Most Cisco devices are well-supported by the standard capabilities of Zenoss. This ZenPack extends those basic capabilities to support modeling and monitoring of characteristics specific to Cisco devices.

27.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.4 or higher
Required ZenPacks	ZenPacks.zenoss.CiscoMonitor

Table 27.1. Cisco Prerequisites

27.3. Enable Monitoring

Follow the steps in this section to configure your Cisco device and Zenoss for monitoring.

27.3.1. Configuring Cisco Devices to Allow SNMP Queries

Configure the Cisco device to allow SNMP queries from the Zenoss server, and send SNMP v1 or SNMP v2 traps to the Zenoss server.

27.3.2. Configuring Zenoss

All Cisco devices must be located in the `/Devices/Network/Cisco` device class.

1. Navigate to the device or device class in the Zenoss interface.
   - If applying changes to a device class:
     a. Select the class in the devices hierarchy.
     b. Click **Details**.
     c. Select Configuration Properties.
   - If applying changes to a device:
     a. Click the device in the device list.
     b. 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.
zSnmpMonitorIgnore	Set to a value of `False`.
zSnmpPort	The default port is 161.

Name	Description
zSnmpVer	Set to a value of `v2c`.

Table 27.2. Cisco Configuration Properties

3. Click Save to save your changes. Zenoss now will collect Cisco device metrics from the configured device or devices.
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.

27.4. Forwarding Syslog Messages to Zenoss

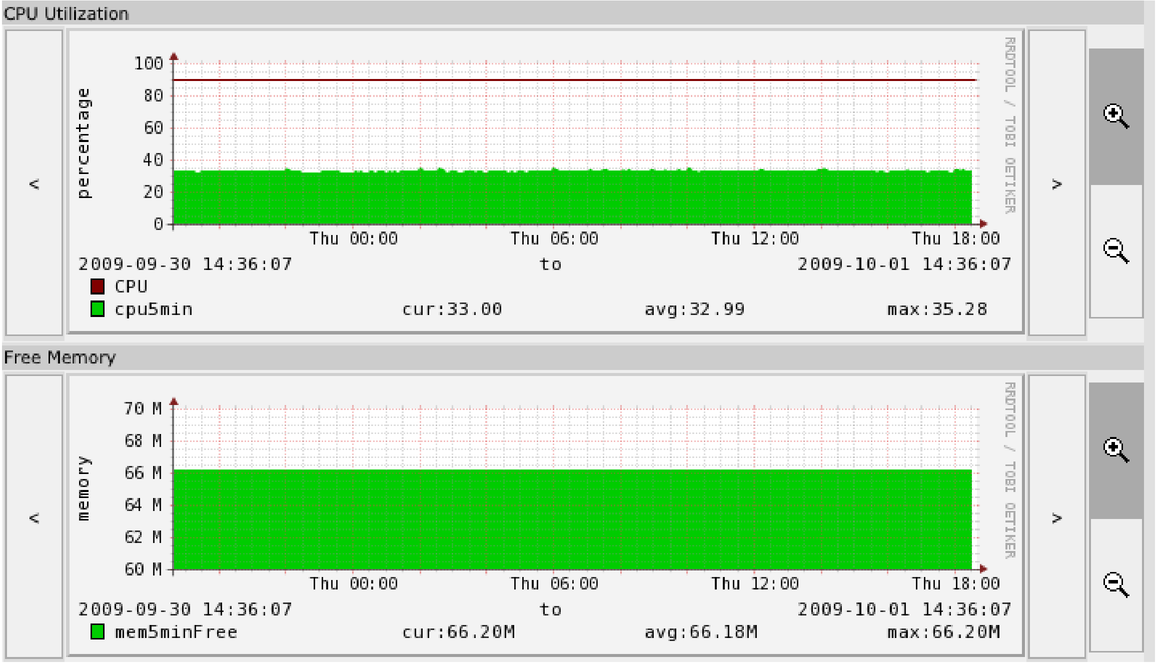
For information about forwarding syslog messages from Cisco IOS routers and CatOS switches into Zenoss, see Appendix C, "Syslog Device Preparation" in *Zenoss Administration*.

27.5. Extended Capabilities for Cisco Devices

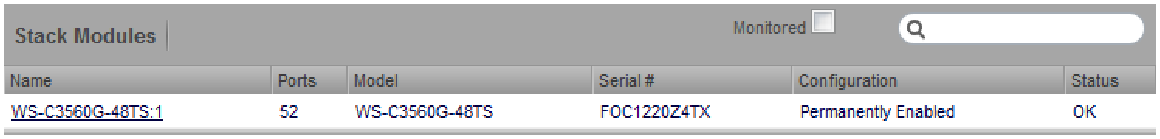
27.5.1. IOS

You should place Cisco devices running IOS in the `/Devices/Network/Cisco` device class. This lets them benefit from these extended monitoring capabilities:

- Modeling of hardware serial number. This information can be found in the details information for Cisco IOS devices.
- Monitoring of CPU and memory utilization. This information can be found on the Graph page for Cisco IOS devices.



- Modeling and monitoring of IP-SLA (RTTMON). This information can be found by navigating to the device, and then selecting Cisco Details from the left panel. menu.
- Modeling of stacked switch modules. This information can be found by navigating to the device, and then selecting Cisco Details from the left panel.



Name	Ports	Model	Serial #	Configuration	Status
WS-C3560G-48TS:1	52	WS-C3560G-48TS	FOC1220Z4TX	Permanently Enabled	OK

27.5.2. CatOS

You should place Cisco Catalyst devices running CatOS in the `/Network/Cisco/CatOS` device class. The only difference in this class is that the CPU and memory performance monitoring is done by using a different configuration. Otherwise, the devices are treated the same as IOS devices.

27.5.3. ASA, FWSM and PIX

You should place Cisco ASA, FWSM, and PIX devices in the `/Network/Cisco/ASA` device class. The only difference in this class is that the CPU and memory performance monitoring is done by using a different configuration. Otherwise, the devices are treated the same as IOS devices.

Zenoss can encounter problems when querying Cisco PIX, ASA, and FWSM devices using SNMP. This is because, by default, Zenoss tries to fit forty requests into a single SNMP packet when using SNMP v2c. This improves performance and reduces network and processing overhead on Zenoss and the monitored device.

Common symptoms of this problem include:

- `DEBUG` level `/Perf/Snmp` events with a summary field of `Error reading value for "???"`
- Missing performance graphs.
- Errors similar to this that appear in the Cisco device log:

```
incoming SNMP request (? bytes) from IP address ?.?.?.? Port ? Interface
    "inside" exceeds data buffer size, discarding this SNMP request.
```

27.5.4. Wireless LAN Controllers

You should place Cisco Wireless LAN Controllers in the `/Network/Cisco/WLC` device class. This lets them benefit from the following extended monitoring capabilities:

- Modeling of hardware model, serial number and operating system. This information can be found on the device Overview page.
- Modeling of individual access points controller by the wireless LAN controller. This information can be found by navigating to the wireless LAN controller device, and then selecting Wireless in the left panel.

27.5.5. ACE Load Balancers

You should place Cisco ACE (Application Control Engine) devices in the `/Network/Cisco/ACE` device class. This lets them benefit from the following extended monitoring capabilities:

- Modeling and monitoring of individual load balanced virtual servers. This information can be found by navigating to the device, and then selecting Cisco Details in the left panel.

27.5.6. Telepresence Codecs

You should place Cisco Telepresence Codec devices in the /Network/Cisco/Codec device class. This lets them benefit from the following extended monitoring capabilities:

- Modeling of hardware model, serial number and operating system. This information can be found on the overview page for Telepresence Codec devices.

- Modeling of all peripherals controlled by the codec. This information can be found by navigating to the Telepresence Code device, and then selecting Telepresence from the left panel.

27.6. Daemons

Type	Name
Modeler	**zenmodeler**
Performance Collector	**zenperfsnmp**

Table 27.3. Daemons

Chapter 28. Cisco UCS

28.1. About

The CiscoUCS ZenPack enables Zenoss to use HTTP to monitor Cisco Unified Computing System (UCS) devices. Using Cisco's UCSTM™ Manager XML API, the system models and monitors devices placed in the `/CiscoUCS` device class.

The Cisco UCS ZenPack provides:

- Fabric interconnect monitoring
- Monitoring of equipment chassis and their compute blades
- Monitoring of service profiles, their compute blade assignments, and links to any other Zenoss device from the UCS service profile on which it is running
- Full monitoring of events generated by the UCS

28.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.5.2 or higher
Required ZenPacks	ZenPacks.zenoss.CiscoUCS, ZenPacks.zenoss.DynamicView

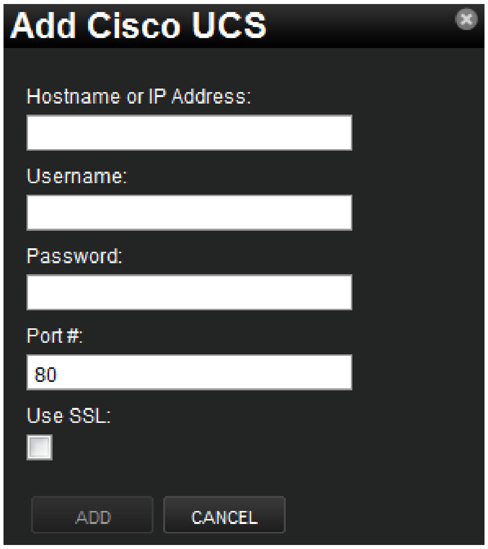
Table 28.1. Cisco UCS Prerequisites

28.3. Adding a Cisco UCS Device for Monitoring

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

1. In the Zenoss interface, navigate to the `/CiscoUCS` device class.
2. From ⊞▾, select Add Cisco UCS.

   The Add Cisco UCS dialog appears.



Add Cisco UCS

Hostname or IP Address:

Username:

Password:

Port #: 80

Use SSL: □

Figure 28.1. Add Cisco UCS Unit

3. Enter information in the dialog:
    - **Hostname or IP Address** - Enter the host name or IP address of the UCS manager.
    - **Username** - Enter the user name of an authorized user.
    - **Password** - Enter the password to the user account.
    - **Port #** - By default, Zenoss assumes a standard HTTP port of 80. Change this value as needed.
4. Click **Add Unit** to begin discovery.

28.4. UCS Monitoring Credentials

These configuration properties are populated automatically if you use the Add Cisco UCS dialog. (See the previous section, Adding a Cisco UCS Device for Monitoring.)

Name	Description
zCiscoUCSManagerUser	Username that will be used to access the Cisco UCS through the UCS Manager.
zCiscoUCSManagerPassword	Password to validate the username.
zCiscoUCSManagerPort	Port number used to monitor the Cisco UCS. The default value is 80.

Table 28.2. Cisco UCS Configuration Properties

28.5. Daemons

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

Table 28.3. Daemons

Chapter 29. Datacenter View

29.1. About

Datacenter View is a visual representation of devices (such as a server or blade and device containers (such as a rack or chassis) in the system. Using 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.

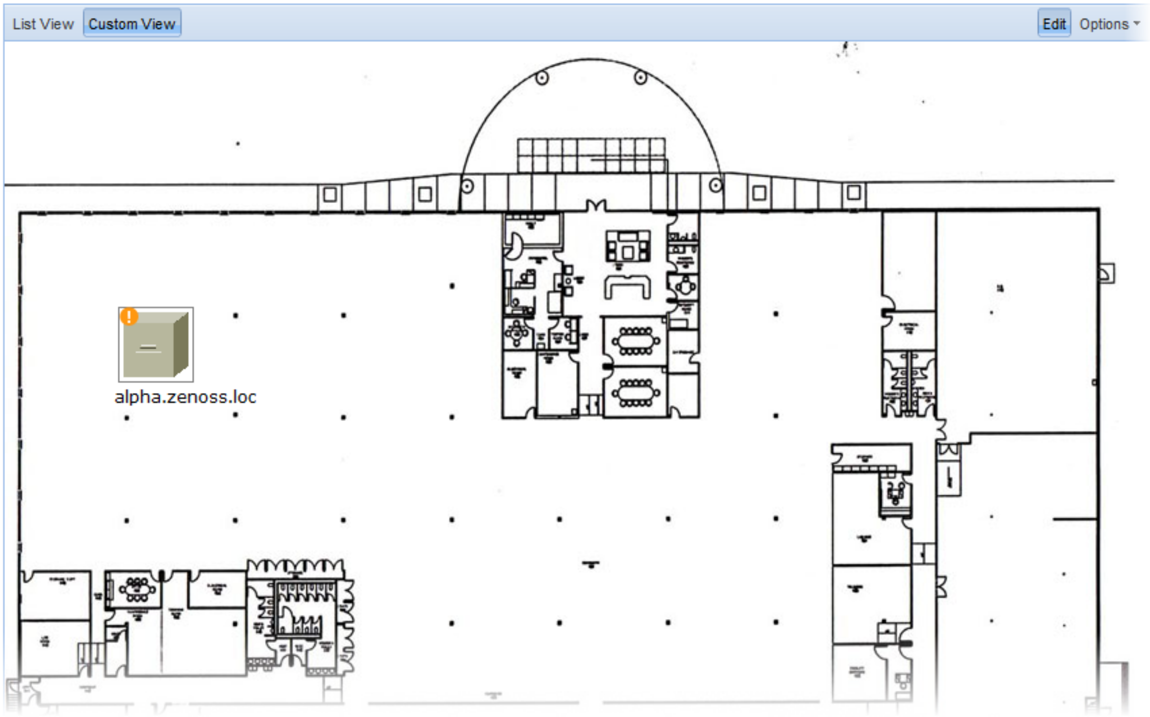


Figure 29.1. Custom View

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.

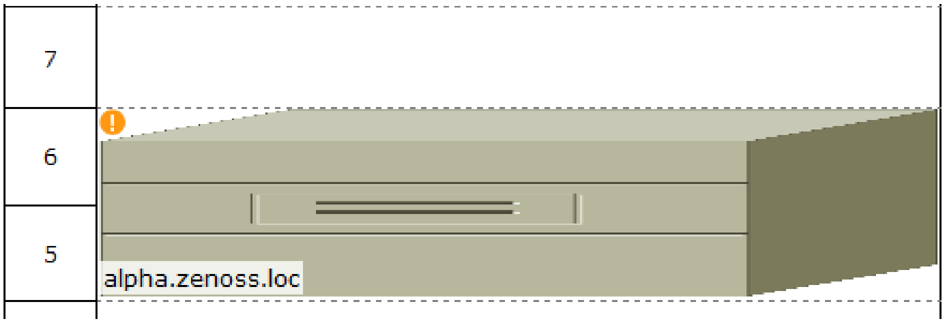


Figure 29.2. Rack View

29.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.5.1 or later
Required ZenPacks	ZenPacks.zenoss.Diagram

Table 29.1. Datacenter View Prerequisites

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 about this view, see the section titled *Activating the Auto-Generated Rack View*.)

29.3. 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 Diagram.

   The List View appears.

   Note

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

29.4. 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

29.4.1. Adding a Background Image to the Custom View

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

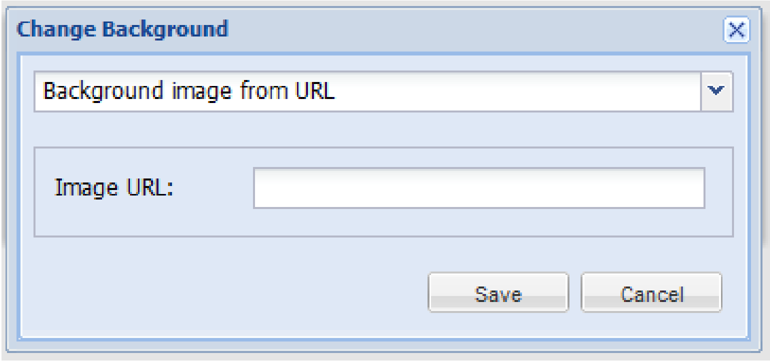
1. From the Datacenter View page (accessed from the Diagram selection), click **Custom View.**
2. Click **Edit** to enable edit mode.

   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.



Change Background

Background image from URL

Image URL:

Figure 29.3. Change Background

29.4.1.1. 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.

29.4.2. 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.

29.4.3. 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.

29.5. 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:

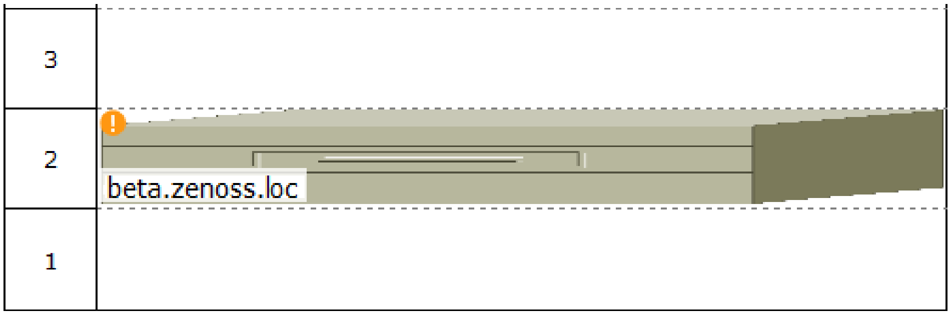
   

   Figure 29.4. Setting Rack Slot Value

   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.

Chapter 30. Device Access Control Lists

30.1. About

The Device Access Control List (ACL) Enterprise ZenPack (ZenDeviceACL) adds fine-grained security controls to Zenoss. You can use this control to limit 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.

30.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.ZenDeviceACL

Table 30.1. Device ACL Prerequisites

30.3. Key Concepts

30.3.1. Permissions and Roles

Actions in Zenoss 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 Zenoss roles, refer to *Zenoss Administration*.

30.3.2. 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. Zenoss 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.

30.3.3. Users and Groups

Users and user groups work exactly as they would normally. See the chapter titled "Managing Users" in *Zenoss Administration* for more information about managing users and groups.

30.3.4. Assigning Administered Object Accessa

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.

30.3.5. Restricted Screen Functionality

30.3.5.1. 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.

30.3.5.2. Device List

The device list is automatically filtered to devices of a restricted user, scoped to accessible devices. There are no menu items available.

30.3.5.3. 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.

30.3.5.4. Reporting

Reports are limited to device reports and performance reports.

30.3.5.5. 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.

30.4. 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.

30.5. Create a Manager Restricted to Specific Devices

Following the previous example:

1. From the user's edit page, change the acltest user’s role to “ZenManager." (You must do this as a user with ZenManager global rights.)
2. Go back to the acltest user's Administered Objects and set the role on the device to ZenManager.
3. As acltest, navigate to the device. You now have access to the Edit page.

30.6. 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.

30.7. 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.

Chapter 31. Distributed Collector

31.1. About

Distributed Collector allows you to deploy additional performance collection and event monitoring daemons to the Zenoss server or other servers. This allows you to:

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

When you first install Distributed Collector, Zenoss is configured with one hub and one collector. A collector is a set of collection daemons, on the Zenoss server or another server, that shares a common configuration. That configuration contains values, such as number of seconds between SNMP collection cycles, default discovery networks, and maximum number of `zenprocess` parallel jobs.

Each collector has its own copy of each of the Zenoss collection daemons. For example, Zenoss 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.

You cannot delete the initial hub and collector set up by Distributed Collector (both named localhost).

31.1.1. Navigating Existing Collectors and Hubs

When you log in as the Zenoss admin user, go to Advanced > Collectors. The Collectors 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. The Daemons selection lists the copy of the `ZenHub` daemon that belongs to the collector. Links adjacent to the daemon name allow you to view its log, and view and edit its configuration. Use the buttons to the right of the daemon name to stop, start, and restart the daemon.

31.1.2. Restrictions and Requirements

- Servers hosting remote hubs or collectors must be the same operating system and hardware architecture as the Zenoss server. For example, if the Zenoss server is running RedHat Enterprise Linux v5 on Intel 32-bit hardware, then hubs and collectors can be deployed only to other RHEL 5 32-bit servers.
- By default, port 8789 must be open so that a distributed collector can communicate with ZenHub. (This can differ if you have configured ZenHub to run on a different port.) For a remote ZenHub, port 3306 most be open for MySQL communications, and port 8100 must be open for ZEO communications.
- You must update all hubs and collectors after performing any of these functions on your master Zenoss server:
    - Upgrade

  - Install patches
  - Install, upgrade, or remove ZenPacks

  To update a hub, select Overview, and then select Update Hub from the Action menu. To update a collector, select it, and then select Update Collector from the Action menu.

- Zenoss is not compatible with Security-Enhanced Linux (SELinux) in enforcing mode. You must disable enforcing mode for all platforms running the Zenoss daemons (Zenoss master, remote hubs, and remote collectors).

  To disable enforcing mode:

  1. Edit the `/etc/linux/config` file.
  2. Set the following line:

     ```
     SELINUX=disabled
     ```

  Note

  You also can disable enforcing mode temporarily (avoiding the need to reboot) with the command:

  ```
  echo 0 > /selinux/enforce
  ```

  For more information about SELinux, browse to http://en.wikipedia.org/wiki/SELinux, or to the SELinux home page at http://www.nsa.gov/research/selinux/index.shtml.

For additional platform-specific information, refer to Section 31.1.5, “Platform Notes”.

31.1.3. Installation Notes

- Make sure the Zenoss server's hostname is a fully qualified domain name.
- Remember that collectors and hubs can be pushed only to servers with identical operating system versions and hardware architecture.
- When installing a remote hub, make sure that Event Manager > hostname has a fully qualified domain name (preferred) or at least a numeric address that can be reached by any server with hubs deployed to it.
- If you have any other firewalls on the Zenoss server, or on servers that host remote collectors or hubs, then you should disable them.

31.1.4. Firewall Notes

Remote hubs need to communicate with the ZEO database on the Zenoss server on port 8100. Hubs also need to communicate with the MySQL server, usually on the Zenoss server (see Event Manager > Hostname), and on the port specified in Event Manager > Port (usually 3306.) Collectors communicate with their hub on the port specified when the hub was created. See the ZenHub Port field on the hub's overview page.

31.1.5. Platform Notes

Software Appliance and Hardware Appliance

- Hubs and collectors can be deployed only to other Zenoss software or hardware appliances.
- You must stop Zenoss on an appliance before deploying a hub or collector to it.
- You must set a password for the root user on an appliance before deploying a hub or collector to it.
- When using appliances for the Zenoss server and remote server the user must shutdown Zenoss on the remote server before creating hub or collectors on it. Otherwise, ZEO, Zope, and the standard Zenoss daemons will run indefinitely on that server and will no longer be controllable via the **zenoss** script.

- Do not use conary to update appliances being used as remote collectors or hubs.

31.1.6. Debugging

Hostname Configuration

The Zenoss server should have a properly configured hostname (preferably a fully qualified domain name). You can check the hostname from the shell:

```
root# hostname
```

You also can check by using the Python function used by Zenoss:

```
root# python -c 'import socket; print socket.gethostname()'
```

31.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.4 or higher
Required ZenPacks	ZenPacks.zenoss.DistributedCollector

Table 31.1. Distributed Collector Prerequisites

31.3. Typical Usage Scenarios for Distributed Monitoring

Typical setup scenarios for using multiple hubs and collectors are:

- ZeoDB - local hub - local collector
- ZeoDB - local hub - remote collector
- ZeoDB - local hub - multiple remote collectors
- ZeoDB - multiple remote hubs - multiple remote collectors

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.

31.3.1. ZeoDB - Local Hub - Local Collector

This setup requires only a single server, and is the most common Zenoss deployment type. You would most likely use this configuration if you need to monitor fewer than 1000 devices, and your master Zenoss server has direct network access to all of the monitored devices.

31.3.2. ZeoDB - 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.

31.3.3. ZeoDB - Local Hub - Multiple Remote Collectors

This is the most common distributed Zenoss configuration. Two reasons you might use this configuration are:

- Scaling Zenoss to monitoring 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.
- Handling differing network security policies. Often, your master Zenoss 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.

31.3.4. ZeoDB - 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.

31.4. Deploying Collectors

Use the information and steps in the following sections to deploy and manage collectors.

Note

Before deploying a remote collector you must set up a remote server. For more information setup tasks, refer to the chapter titled "Installing Distributed Collectors" in *Zenoss Installation* for Enterprise.

31.4.1. Prerequisite Tasks

All prerequisite tasks and conditions required to install Zenoss are also required by the machine that will be the remote collector. Refer to *Zenoss Installation* for Enterprise for specific procedures to satisfy these conditions.

By default, only local access to the ZEO database is configured. Before adding a remote hub, you must edit the `$ZENHOME/etc/zeo.conf` file to allow remote access.

In the file, change the line:

```
address localhost:8100
```

to

```
address 8100
```

31.4.2. Adding Collectors

To add a collector to a hub:

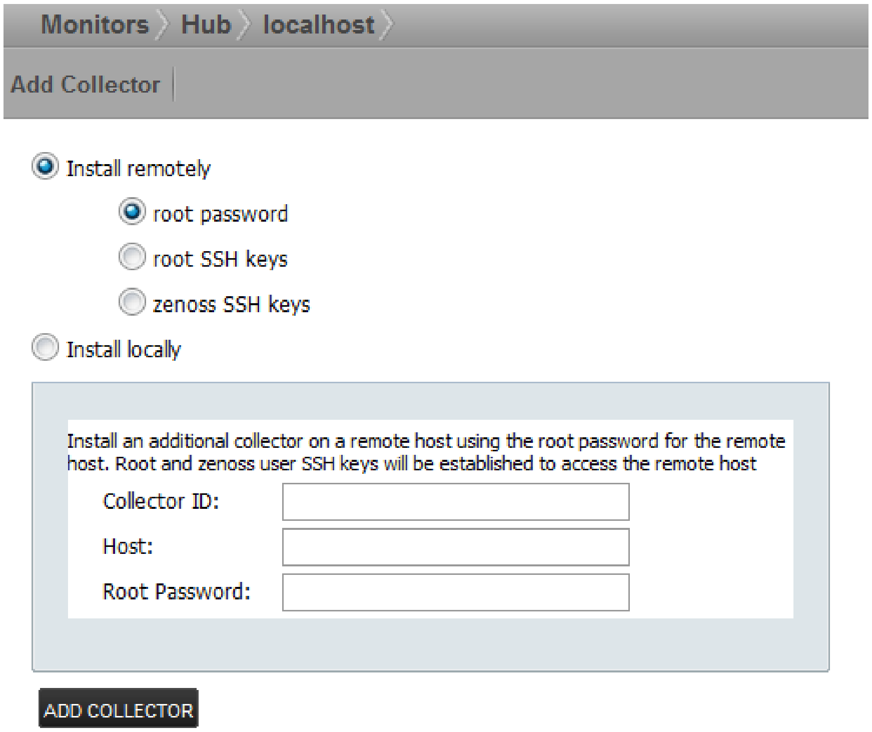
1. Navigate to the Hub Overview page.
2. Select Add Collector from the Zenoss Collectors Action menu.

   The Add Collector page appears.

31.4.2.1. Install Remotely (Root Password)

To install a remote collector, using a root password for access to the remote host:

1. Select the Install remotely option.
2. Select the root password option.



● Install remotely

● root password

○ root SSH keys

○ zenoss SSH keys

○ Install locally

Install an additional collector on a remote host using the root password for the remote host. Root and zenoss user SSH keys will be established to access the remote host

Collector ID:	
Host:	
Root Password:	

Figure 31.1. Install Remote Collector (Root Password)

3. Enter or change setup details:

Field Name	Description
Collector ID	Enter the name for the collector as it will be identified in Zenoss. This name will be used to prefix the Zenoss control scripts on the collector. If the ID is `coll1`, then scripts will be named `coll1_zenperfsnmp`.
Host	Enter the name of the host for the collector. This must be a fully qualified domain name, IP address, or resolvable hostname.
Root Password	Enter the password for the root user on the Host. The root password is not stored; it is used to configure a pre-shared key between the main Zenoss server and the remote collector.

Table 31.2. Add New Collector Fields

Note

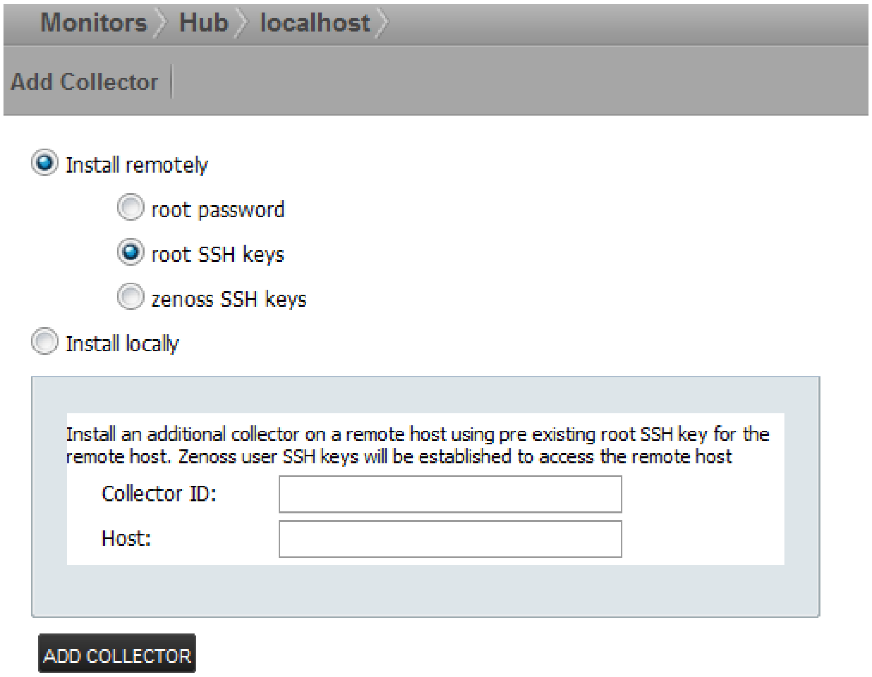
If you are creating another collector on the Zenoss server, enter the `localhost` rather than the IP address of the Zenoss server.

4. Click **Add Collector**. The system displays log output from the creation of the new collector. When fully configured (this may require several minutes), click the link at the bottom of the page to go to the overview page for the new collector.

31.4.2.2. Install Remotely (Root SSH Keys)

To install a remote collector, using existing root SSH keys for access to the remote host:

1. Select the Install remotely option.
2. Select the root SSH keys option.



● Install remotely

○ root password

● root SSH keys

○ zenoss SSH keys

○ Install locally

Install an additional collector on a remote host using pre existing root SSH key for the remote host. Zenoss user SSH keys will be established to access the remote host

Collector ID:

Host:

Figure 31.2. Install Remote Collector (Root SSH Keys)

3. Enter or change setup details:

Field Name	Description
Collector ID	Enter the name for the collector as it will be identified in Zenoss. This name will be used to prefix the Zenoss control scripts on the collector. If the ID is `coll1`, then scripts will be named `coll1_zenperfsnmp`.
Host	Enter the name of the host for the collector. This must be a fully qualified domain name, IP address, or resolvable hostname.

Table 31.3. Add New Collector Fields

> **Note**
>
> If you are creating another collector on the Zenoss server, enter the `localhost` rather than the IP address of the Zenoss server.

4. Click **Add Collector**. The system displays log output from the creation of the new collector. When fully configured (this may require several minutes), click the link at the bottom of the page to go to the overview page for the new collector.

31.4.2.3. Install Remotely (Zenoss SSH Keys)

If you choose to set up a collector using Zenoss SSH keys, Zenoss will attempt to install by using the zenoss user. To successfully install a collector using these keys (without root access), these prerequisite conditions must be met:

- zenoss user SSH keys must be set up between the Zenoss server and the target.
- You must be running the RPM distribution of Zenoss.
- Zenoss core RPM must be installed on the target (remote) machine.

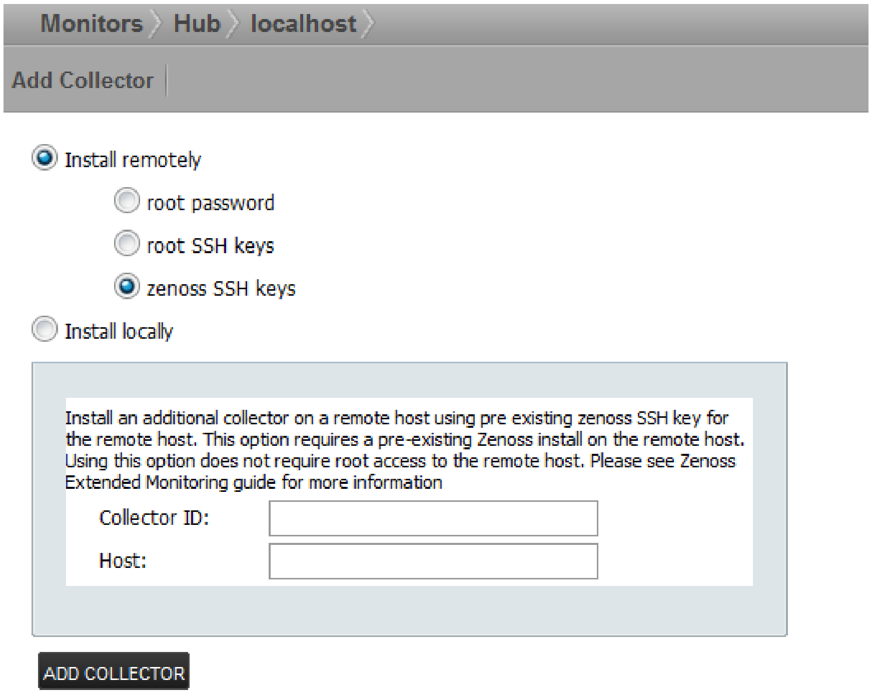
Tip: When installing the Zenoss RPM on the remote machine, **do not start** Zenoss.

Follow these steps to install a remote collector, using Zenoss SSH keys for access to the remote host.

Note

For detailed steps for creating SSH keys, see the section titled "Setting Up SSH Keys for Distributed Collector."

1. Select the Install remotely option.
2. Select the zenoss SSH Keys option.



● Install remotely

○ root password

○ root SSH keys

● zenoss SSH keys

○ Install locally

Install an additional collector on a remote host using pre existing zenoss SSH key for the remote host. This option requires a pre-existing Zenoss install on the remote host. Using this option does not require root access to the remote host. Please see Zenoss Extended Monitoring guide for more information

Collector ID:

Host:

Figure 31.3. Install Remote Collector (Zenoss SSH Keys)

3. Enter or change setup details:

Field Name	Description
Collector ID	Enter the name for the collector as it will be identified in Zenoss. This name will be used to prefix the Zenoss control scripts on the collector. If the ID is `coll1`, then scripts will be named `coll1_zenperfsnmp`.

Field Name	Description
Host	Enter the name of the host for the collector. This must be a fully qualified domain name, IP address, or resolvable hostname.

Table 31.4. Add New Collector Fields

Note

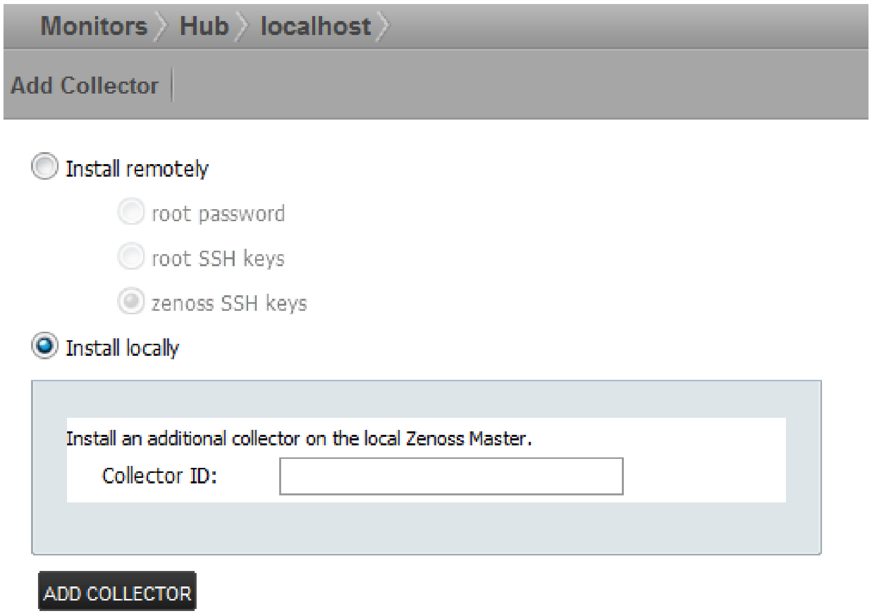
If you are creating another collector on the Zenoss server, enter the `localhost` rather than the IP address of the Zenoss server.

4. Click **Add Collector**. The system displays log output from the creation of the new collector. When fully configured (this may require several minutes), click the link at the bottom of the page to go to the overview page for the new collector.

31.4.2.4. Install Locally

Follow these steps to install a local collector:

1. Select the Install locally option.



Install an additional collector on the local Zenoss Master.

Collector ID:

Figure 31.4. Install Remote Collector (Zenoss SSH Keys)

2. Enter or change setup details:

Field Name	Description
Collector ID	Enter the name for the collector as it will be identified in Zenoss. This name will be used to prefix the Zenoss control scripts on the collector. If the ID is `coll1`, then scripts will be named `coll1_zenperfsnmp`.

Table 31.5. Add New Collector Fields

3. Click **Add Collector**. The system displays log output from the creation of the new collector. When fully configured (this may require several minutes), click the link at the bottom of the page to go to the overview page for the new collector.

31.4.3. 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 Zenoss Collectors, select the collector you want to delete. From the Action menu, select Delete Collector.

When you delete collectors using this Zenoss 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.

31.4.4. Updating a Hub or Collector

Warning

Any time you update your version of Zenoss or install additional ZenPacks, you must update any hubs or collectors.

To update a hub or collector, navigate to the Overview page for the hub or collector, and then choose Update Hub or Update Collector from the Action menu. This copies the most recent Zenoss code and ZenPacks to the server and restarts the daemons running there.

31.4.5. Backing Up Remote Collectors

Zenoss 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.

Note

Zenoss recommends that you avoid performing backups directly to NFS file systems. Because `zenbackup` must restart Zenoss after a backup, a bad connection to an NFS server can prevent the remote collector from starting.

31.5. Adding Devices to Collectors

Adding devices to collectors occurs when you add the device to Zenoss.

1. Select Add a Single Device from (Add Device).

   The Add a Single Device page appears.

2. From the Collector list of options, select the collector you want to use to collect data for the device.

   After you select the collector, the device appears in the Devices list, located at the bottom of the collector overview page.

31.5.1. 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**.

   Zenoss moves the devices to the selected collector.

   Note

   When a device is moved between collectors, the performance data is not moved. As a result, historical data for the device may not appear in reports and graphs.

31.6. Managing the Collector Daemons

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

31.7. Deploying Hubs

In addition to collectors, Distributed Collector allows you to set up new hubs. A hub represents an instance of the zenhub daemon, which is the daemon through which all collector daemons communicate with the object database and event database. All collectors must belong to exactly one hub; however, a hub may 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 will benefit from multiple hubs.

Hubs are used to manage configuration data and pass it to the collectors. Hubs also take data from the collectors and pass it to the ZeoDB. More 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.

31.7.1. Configuring MySQL for Remote Hubs

Hubs on remote servers need access to the MySQL events database. This setting is the Hostname field in the Connection Information section of the Event Manager page. By default this is set to localhost, but will not work for remote hubs. Distributed collector attempts to set this field to the fully qualified domain name of the Zenoss server when it is installed. If remote hubs appear to be having trouble connecting to MySQL or sending events, then check the value in this field to make sure it can be reached from the server the hub is on.

Another aspect of remote hubs connecting to MySQL is privileges. For a hub to connect to the events database, the user specified in the User Name field in Event Settings must be granted privileges to connect to MySQL from the remote server. Distributed Collector attempts to grant these privileges any time a remote hub is created or updated. If a remote hub is logging error messages that indicate it is not allowed to connect to MySQL from the given host, then these privileges are likely not set up correctly. Granting of these privileges requires a fully qualified domain name for the remote server.

Before adding a hub, ensure MySQL grants and permissions are set correctly.

The zenoss user needs the following privileges set to see if a remote connection is possible:

```
GRANT SELECT on mysql.user to zenoss@localhost IDENTIFIED BY "zenoss";
FLUSH PRIVILEGES;
```

In addition, a zenoss MySQL user is needed that can access the database by using the fully qualified domain name of the zenoss installation:

```
GRANT ALL PRIVILEGES ON events.* to zenoss@'<FQDN>' IDENTIFIED BY "zenoss";
GRANT SELECT on mysql.user to zenoss@'<FQDN>' IDENTIFIED BY "zenoss";
FLUSH PRIVILEGES;
```

When you add the remote hub, you will see an error that indicates how to add a remote MySQL user for the hub to be installed. To resolve this issue, do one of the following:

- Open remote privileges to the MySQL database with:

  ```
  GRANT ALL PRIVILEGES ON events.* to zenoss@'%' IDENTIFIED BY "zenoss";
  FLUSH PRIVILEGES;
  ```

  OR

- Add a zenoss MySQL user for each remote hub:

  ```
  GRANT ALL PRIVILEGES ON events.* to zenoss@'<ZENHUB FQDN>' IDENTIFIED BY "zenoss";
  FLUSH PRIVILEGES;
  ```

31.7.2. Add a Hub

When installing a remote hub, you can select one of several options, using:

- Root password to the remote host
- Pre-existing root SSH keys
- Zenoss SSH keys (use only for RPM installations)

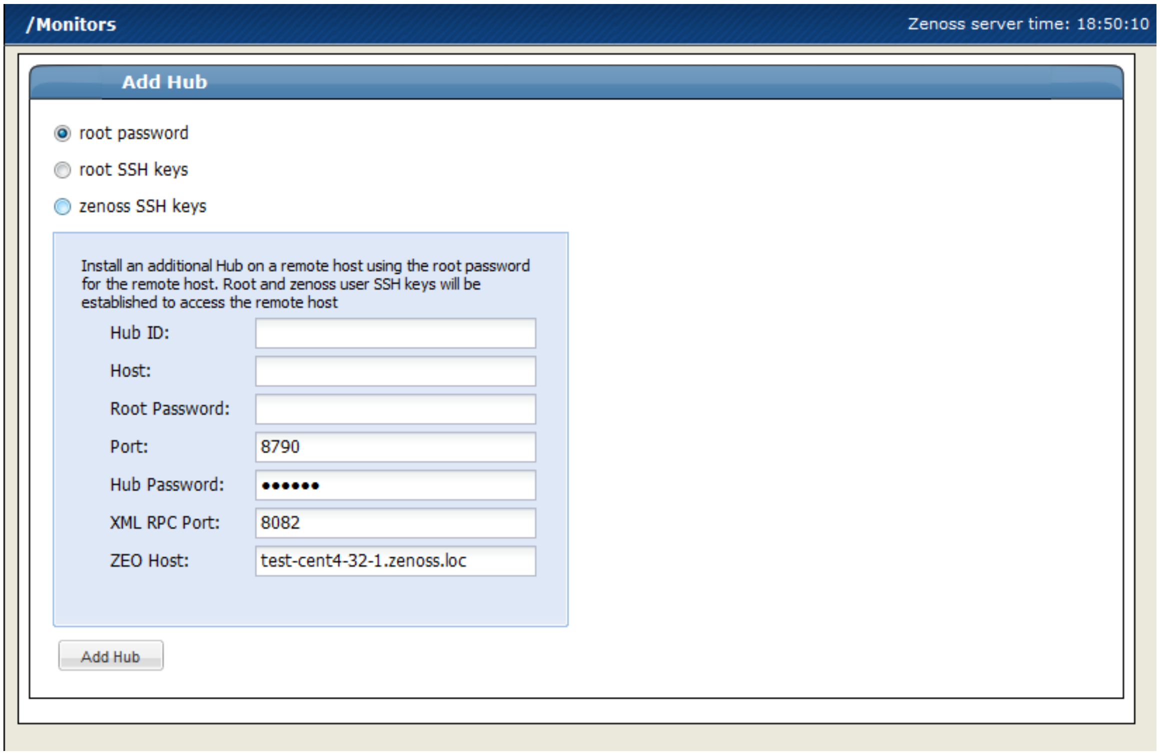
To add a hub, from the main Collectors page, select Add Hub from the Action menu.

The Add Hub page appears.

31.7.2.1. Install Remotely (Root Password)

To install a remote hub, using a root password for access to the remote host:

1. Select the root password option.



● root password

○ root SSH keys

○ zenoss SSH keys

Install an additional Hub on a remote host using the root password for the remote host. Root and zenoss user SSH keys will be established to access the remote host

Hub ID:	
Host:	
Root Password:	
Port:	8790
Hub Password:	••••••
XML RPC Port:	8082
ZEO Host:	test-cent4-32-1.zenoss.loc

Figure 31.5. Install Remote Hub (Root Password)

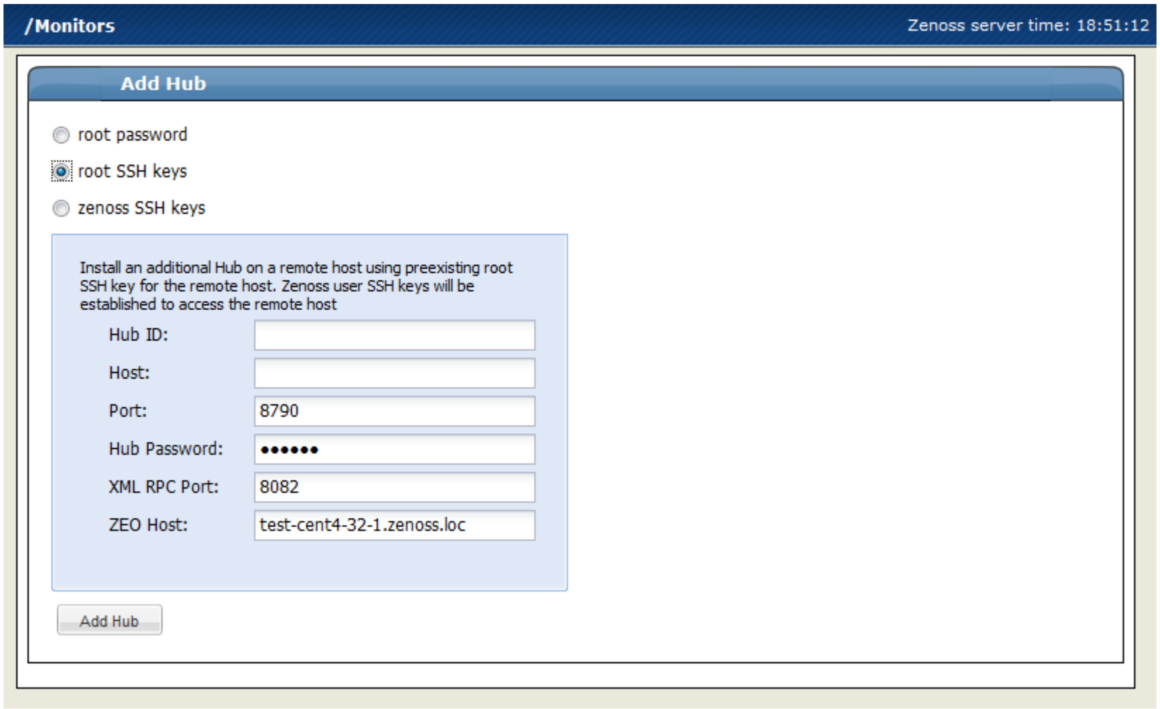
2. Enter or change setup details:
   - **Hub ID** - Enter a name for the new hub. The name can be any unique combination of letters, digits, and dashes.
   - **Host** - Enter the fully qualified domain name, IP address, or resolvable hostname of the server on which the new hub will run.
   - **Root Password** - Enter the root user password for the server you specified in the Host field.
   - **Port** - Enter the port number on which the hub should listen for collectors. The default port is 8790.
   - **Hub Password** - Enter the hub password that the collectors will use to log in to this hub. The default password is "zenoss."
   - **XML RPC Port** - Specify the port on which the hub should listen for xml-rpc requests from the collectors or other API clients.
   - **ZEO Host** - Specify the server hosting the ZEO database (the object database). In most cases, this is the IP address or hostname of the main Zenoss server.
3. Click **Add Hub**.

   The system displays log output from the creation of the new hub. When fully configured (this may require several minutes), click the link at the bottom of the page to go to the overview page for the new hub.

31.7.2.2. Install Remotely (Root SSH Keys)

To install a remote hub, using existing root SSH keys for access to the remote host:

1. Select the root SSH keys option.



○ root password
● root SSH keys
○ zenoss SSH keys

Install an additional Hub on a remote host using preexisting root SSH key for the remote host. Zenoss user SSH keys will be established to access the remote host

Hub ID:
Host:
Port: 8790
Hub Password: ••••••
XML RPC Port: 8082
ZEO Host: test-cent4-32-1.zenoss.loc

Figure 31.6. Install Remote Hub (Root SSH Keys)

2. Enter or change setup details:
   - **Hub ID** - Enter a name for the new hub. The name can be any unique combination of letters, digits, and dashes.

- **Host** - Enter the fully qualified domain name, IP address, or resolvable hostname of the server on which the new hub will run.
- **Port** - Enter the port number on which the hub should listen for collectors. The default port is 8790.
- **Hub Password** - Enter the hub password that the collectors will use to log in to this hub. The default password is "zenoss."
- **XML RPC Port** - Specify the port on which the hub should listen for xml-rpc requests from the collectors or other API clients.
- **ZEO Host** - Specify the server hosting the ZEO database (the object database). In most cases, this is the IP address or hostname of the main Zenoss server.

3. Click **Add Hub**.

   The system displays log output from the creation of the new hub. When fully configured (this may require several minutes), click the link at the bottom of the page to go to the overview page for the new hub.

31.7.2.3. Install Remotely (Zenoss SSH Keys)

If you choose to set up a hub using Zenoss SSH keys, Zenoss will attempt to install by using the zenoss user. To successfully install a hub using these keys (without root access), these prerequisite conditions must be met:

- zenoss user SSH keys must be set up between the Zenoss server and the target. The target must have a zenoss user.
- ZENHOME directory must be present on the remote machine.
- zensocket must be present on the remote machine, and the setuid bits must be set.

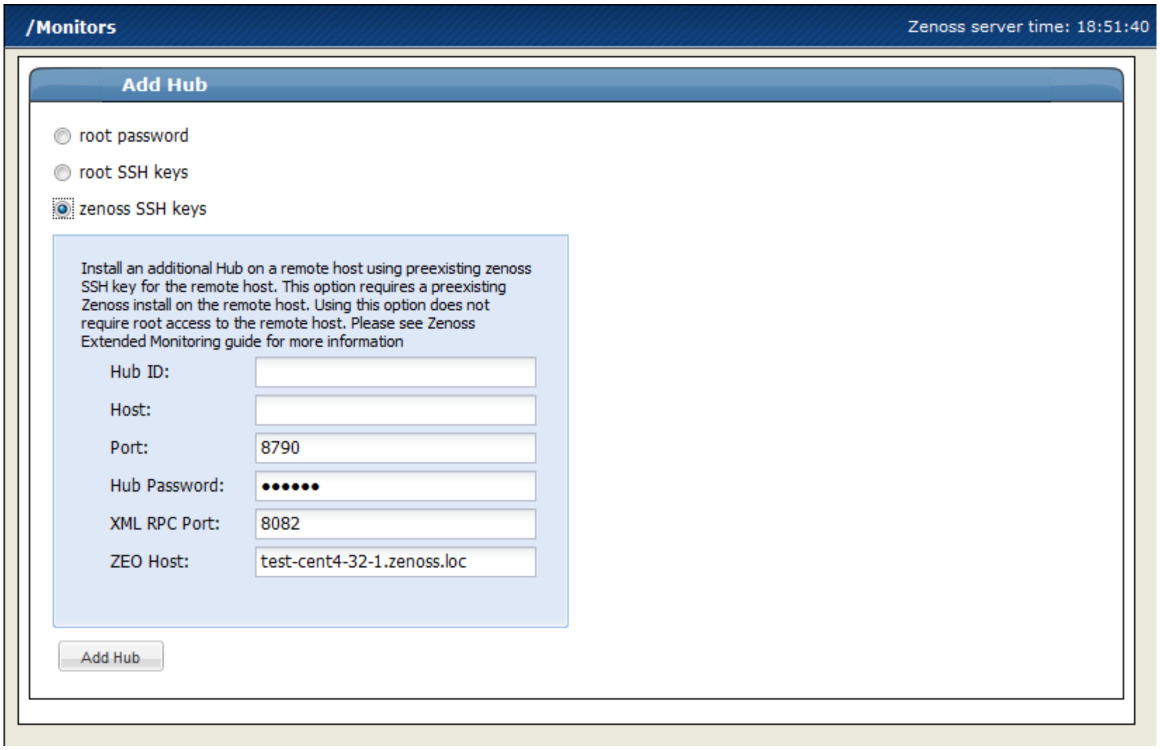
Tip: The best way to meet the prerequisite conditions is to install the Zenoss RPM on the remote machine. After installation, **do not start** Zenoss.

Follow these steps to install a remote hub, using Zenoss SSH keys for access to the remote host.

Note

For detailed steps for creating SSH keys, see the section titled "Setting Up SSH Keys for Distributed Collector."

1. Select the zenoss SSH keys option.



○ root password

○ root SSH keys

● zenoss SSH keys

Install an additional Hub on a remote host using preexisting zenoss SSH key for the remote host. This option requires a preexisting Zenoss install on the remote host. Using this option does not require root access to the remote host. Please see Zenoss Extended Monitoring guide for more information

Hub ID:	
Host:	
Port:	8790
Hub Password:	••••••
XML RPC Port:	8082
ZEO Host:	test-cent4-32-1.zenoss.loc

Figure 31.7. Install Remote Hub (Zenoss SSH Keys)

2. Enter or change setup details:

   - **Hub ID** - Enter a name for the new hub. The name can be any unique combination of letters, digits, and dashes.
   - **Host** - Enter the fully qualified domain name, IP address, or resolvable hostname of the server on which the new hub will run.
   - **Port** - Enter the port number on which the hub should listen for collectors. The default port is 8790.
   - **Hub Password** - Enter the hub password that the collectors will use to log in to this hub. The default password is "zenoss."
   - **XML RPC Port** - Specify the port on which the hub should listen for xml-rpc requests from the collectors or other API clients.
   - **ZEO Host** - Specify the server hosting the ZEO database (the object database). In most cases, this is the IP address or hostname of the main Zenoss server.

3. Click **Add Hub**.

   The system displays log output from the creation of the new hub. When fully configured (this may require several minutes), click the link at the bottom of the page to go to the overview page for the new hub.

31.7.3. Setting Up SSH Keys for Distributed Collector

Follow these instructions to create SSH keys for use when setting up hubs and collectors.

These instructions assume you are using openssh. For more information, refer to the ssh-keygen man pages.

1. Use the following commands to generate an openssh RSA key pair for the zenoss user:

```
mkdir $HOME/.ssh
```

```
ssh-keygen -t rsa -f $HOME/.ssh/id_rsa -P "
```

2. Lock down the key pair:

   ```
   chmod 700 $HOME/.ssh
   chmod go-rwx $HOME/.ssh/*
   ```

3. Copy the generated public key `$HOME/.ssh/id_rsa.pub` file to the remote machine. On the remote machine, add the public key to the `authorized_keys` file in the account the user wants to log in to by using the SSH key.

   a. If `$HOME/.ssh` does not exist on the target machine, then create it with these commands:

      ```
      mkdir ~/.ssh
      chmod 700 ~/.ssh
      ```

   b. Add the key:

      ```
      cat id_rsa.pub >> $HOME/.ssh/authorized_keys
      chmod 600 $HOME/.ssh/authorized_keys
      ```

Note

You cannot use keys with a pass phrase with Zenoss.

Chapter 32. Dynamic Service View

32.1. About

Dynamic Service View ("dynamic view") is a 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.

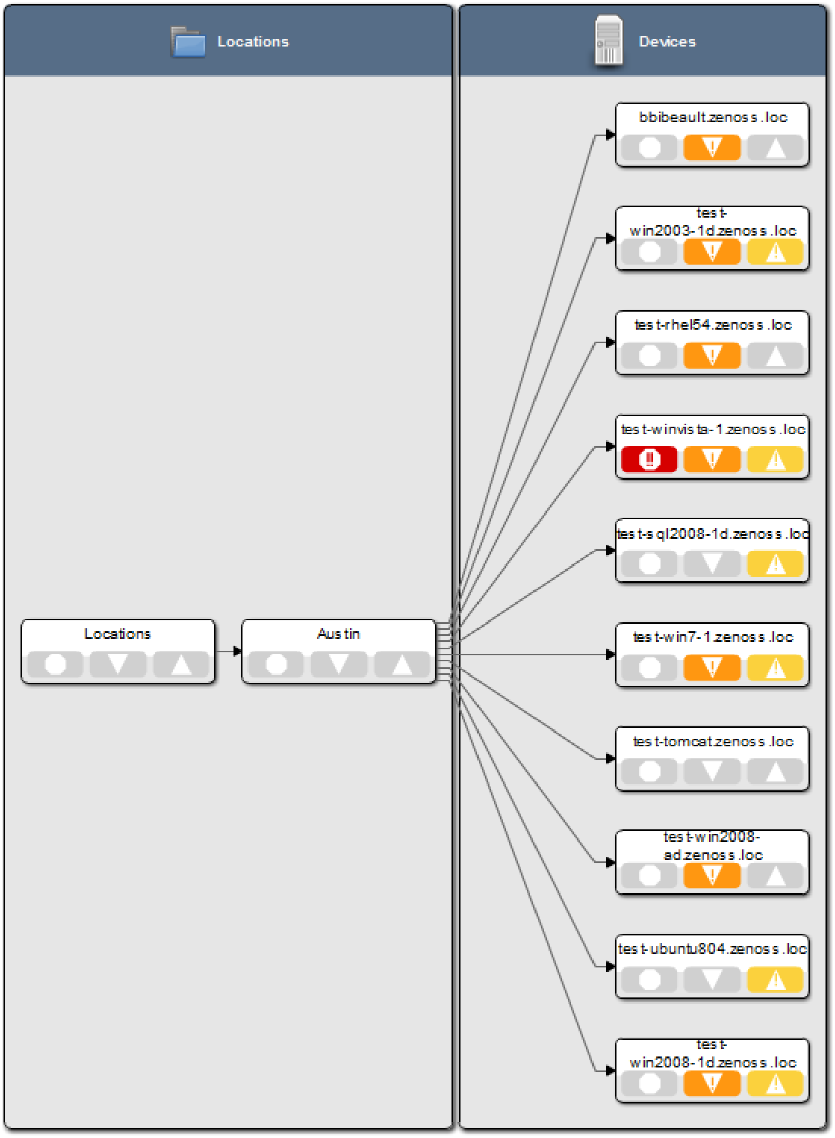
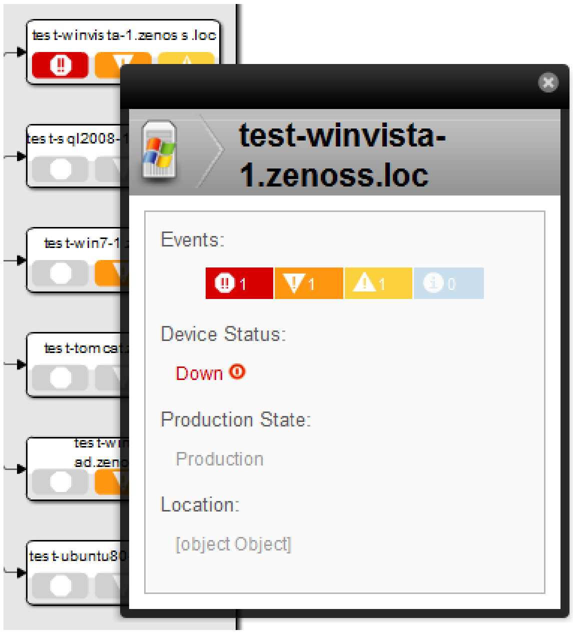

Locations
Devices
bbibeault.zenoss.loc
test-win2003-1d.zenoss.loc
test-rhel54.zenoss.loc
test-winvista-1.zenoss.loc
test-sql2008-1d.zenoss.loc
Locations
Austin
test-win7-1.zenoss.loc
test-tomcat.zenoss.loc
test-win2008-ad.zenoss.loc
test-ubuntu804.zenoss.loc
test-win2008-1d.zenoss.loc

Figure 32.1. 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.



test-winvista-1.zenoss.loc

Events:

1 1 1 0

Device Status:

Down

Production State:

Production

Location:

[object Object]

Figure 32.2. 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.

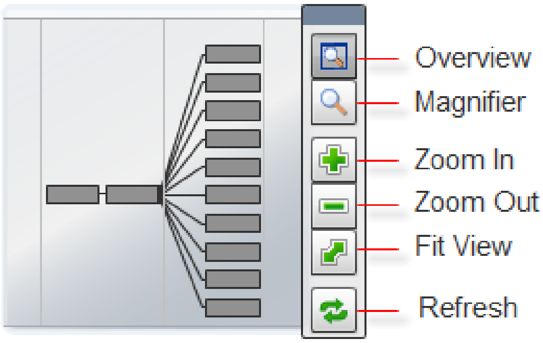


Figure 32.3. Dynamic View: View Controls

32.1.1. 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.

32.1.2. 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.

32.1.2.1. 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.

32.1.2.2. 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

32.1.2.3. 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.

32.2. Prerequisites

Prerequisite	**Restriction**
Zenoss Version	Zenoss Version 3.0 or higher
Required ZenPacks	ZenPacks.zenoss.DynamicView
Other	Oracle/Sun JRE 1.5 or later, Flash-enabled Web browser

Table 32.1. Prerequisites

32.3. Enabling

After installing the DynamicView ZenPack, you must restart the system. The `zenjserver` daemon must be running for dynamic views to be visible.

32.4. Daemons

Type	**Name**
Display	**zenjserver**

Table 32.2. Daemons

Chapter 33. Enterprise Collector

33.1. About

The Zenoss Enterprise Collector ZenPack allows collector daemons to start and monitor devices, even if a connection to ZenHub is not available when the daemon starts.

Enterprise Collector enables configuration caching for these collector daemons:

- `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 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 to the system.

33.2. Prerequisites

Prerequisite	**Restriction**
Zenoss Version	Zenoss Version 2.5 or higher

Table 33.1. Enterprise Collector Prerequisites

33.3. Enabling Enterprise Collector

After installing the Enterprise Collector ZenPack, restart Zenoss and all Zenoss daemons (including `zenhub`).

Chapter 34. Enterprise Linux

34.1. About

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

34.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.4 or higher
Required ZenPacks	ZenPacks.zenoss.LinuxMonitor, ZenPacks.zenoss.EnterpriseLinux

Table 34.1. Enterprise Linux Prerequisites

Note

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

34.3. Add a Linux Server

The following procedure assumes that the credentials have been set.

1. From Infrastructure > Devices, Select Add a Single Device from (Add Device).
2. Enter the following 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 (see Section 34.4, “Set Linux Server Monitoring Credentials”) and then manually model the device.

Table 34.2. Adding Linux Device Information

3. Click **Add**.

34.4. Set Linux Server Monitoring Credentials

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

Tip

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 Zenoss interface.
   - If applying changes to a device class:
     a. Select the class in the devices hierarchy.
     b. Click **Details**.
     c. Select Configuration Properties.
   - If applying changes to a device:
     a. Click the device in the device list.
     b. Select Configuration Properties.
2. Verify the credentials for the service account to access the service.

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

Table 34.3. Linux Configuration Properties

3. Click Save to save your changes.

34.5. 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 Zenoss interface.
   - If applying changes to a device class:
     a. Select the class in the devices hierarchy.
     b. Click **Details**.
     c. Select Configuration Properties.
   - If applying changes to a device:
     a. Click the device in the device list.
     b. Select Configuration Properties.
2. Apply an appropriate value for the maximum number of sessions.

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

Table 34.4. Concurrent SSH Configuration Properties

3. Click Save to save your changes.

34.6. 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 in order to return results from the commands. In order 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 Zenoss interface.
   - If applying changes to a device class:
     a. Select the class in the devices hierarchy.
     b. Click **Details**.
     c. Select Configuration Properties.
   - If applying changes to a device:
     a. Click the device in the device list.
     b. Select Configuration Properties.
2. Apply an appropriate value for the command timeout.

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

Table 34.5. SSH Timeout Configuration Properties

3. Click Save to save your changes.

34.7. 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 on working with Zenoss plugins, refer to *Zenoss Administration*.

34.8. Daemons

Type	**Name**
Modeler	**zenmodeler**
Performance Collector	**zencommand**

Table 34.6. Daemons

Chapter 35. Enterprise Reports

35.1. About

The EnterpriseReports ZenPack adds new reports to the standard Zenoss reports. Available reports include:

- 95th Percentile
- Alert Rule Email Addresses
- Defined Thresholds
- Event Time to Resolution
- Interface Volume
- Maintenance Windows
- Organizer Availability
- User Event Activity
- Users Group Membership

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

35.1.1. 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.

35.1.2. Alert Rule Email Addresses

The Alert Rule Email Addresses report displays all alert rules and the email addresses to which alerts are sent.

This report is useful when reviewing which users receive certain types of system alerts.

35.1.3. 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.

35.1.4. 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.

35.1.5. 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.

35.1.6. 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.

35.1.7. 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.

35.1.8. 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.

35.1.9. Users Group Membership

Shows all users and the groups to which they belong.

35.2. Viewing Enterprise Reports

After installing the EnterpriseReports ZenPack, you can access Enterprise reports. From the Zenoss interface, select Reports from the navigation bar.

35.3. Prerequisites

Prerequisite	Restriction
Zenoss version	Zenoss version 2.2 or higher
Zenoss Product	Zenoss Enterprise
Required ZenPacks	ZenPacks.zenoss.EnterpriseReports

Table 35.1. Enterprise Reports Prerequisites

Chapter 36. Enterprise Security

36.1. About

The EnterpriseSecurity ZenPack enhances Zenoss security by enabling password encryption. Zenoss 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.

36.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.5 or higher
Required ZenPacks	EnterpriseSecurity

Table 36.1. Enterprise Security Prerequisites

36.3. Enabling Password Encryption

To enable password encryption, install the ZenPack. No other action is required to enable this feature. After ZenPack installation, password encryption is always enabled.

To test that password encryption is functioning correctly, use `grep` 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 Zenoss server:

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

If the Enterprise Security ZenPack is installed, this command will not return results.

Chapter 37. Foundry Device

37.1. About

The FoundryMonitor 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`.

37.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.4 or higher
Required ZenPacks	ZenPacks.zenoss.FoundryMonitor

Table 37.1. Foundry Prerequisites

37.3. Configuring Zenoss

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

Follow these steps to configure Zenoss:

1. Navigate to the device or device class in the Zenoss interface.
   - If applying changes to a device class:
     a. Select the class in the devices hierarchy.
     b. Click **Details**.
     c. Select Configuration Properties.
   - If applying changes to a device:
     a. Click the device in the device list.
     b. 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.
zSnmpMonitorIgnore	Set to a value of `False`.
zSnmpPort	The default port is 161.

Name	Description
zSnmpVer	Set to a value of `v2c`.

Table 37.2. Foundry Configuration Properties

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

37.4. Daemons

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

Table 37.3. Daemons

Chapter 38. Hewlett Packard UNIX

38.1. About

The HpuxMonitor ZenPack enables Zenoss to use Secure Shell (SSH) to monitor Hewlett Packard UNIX (HP-UX) hosts. The system models and monitors devices placed in the `/Server/SSH/HP-UX` device class by running commands and parsing the output. Parsing of command output is performed on the system server (if using a local collector) or on a distributed collector. The account used to monitor the device requires root access or special privileges to access `/usr/bin/adb`.

The HpuxMonitor 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)

38.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Version 2.5 or higher
Required ZenPacks	ZenPacks.zenoss.HpuxMonitor
Supported HP-UX Releases	HP-UX 11.00

Table 38.1. HP-UX Prerequisites

Note

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

38.3. Limitations

This ZenPack has not been tested on Itanium systems.

38.4. 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 (Add Device).
2. Enter the following information:

Name	Description
Name or IP	HP-UX host to model

Name	Description
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 (see Section 38.5, “Set HP-UX Server Monitoring Credentials”), and then manually model the device.

Table 38.2. Adding HP-UX Device Information

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

38.5. 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.

38.5.1. 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:

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

Table 38.3. HP-UX Configuration Properties

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

38.5.2. 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.

38.6. 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 Zenoss interface.
   - If applying changes to a device class:
     a. Select the class in the devices hierarchy.
     b. Click **Details**.
     c. Select Configuration Properties.
   - If applying changes to a device:
     a. Click the device in the device list.
     b. Select Configuration Properties.
2. Apply an appropriate value for the maximum number of sessions.

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.

Table 38.4. Concurrent SSH Configuration Properties

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

38.7. 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 Zenoss interface.
   - If applying changes to a device class:
     a. Select the class in the devices hierarchy.
     b. Click **Details**.
     c. Select Configuration Properties.
   - If applying changes to a device:
     a. Click the device in the device list.
     b. Select Configuration Properties.
2. Apply an appropriate value for the command timeout.

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

Table 38.5. SSH Timeout Configuration Properties

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

38.8. Daemons

Type	**Name**
Modeler	**zenmodeler**
Performance Collector	**zencommand**

Table 38.6. Daemons

Chapter 39. JBoss Application Server

39.1. About

the JBossMonitor ZenPack that system administrators to monitor JBoss Application Servers. JBossMonitor 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.

The 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 that are collected in JBossMonitor represent combinations of individual component metrics. For example, the Thread Pool metric represents all threads in all pools. It is possible to configure JBossMonitor 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 JBossMonitor 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

39.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.ZenJMX, ZenPacks.zenoss.JBossMonitor

Table 39.1. JBoss Prerequisites

39.3. Enable Monitoring

39.3.1. 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.

39.3.2. Configuring Zenoss

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

> **Note**
>
> The **zenjmx** daemon must be configured and running. See Section 10.2.1, “Sun Java Runtime Environment (JRE)” for more information about configuring the **zenjmx** daemon with the Sun JRE tools.

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

   b. Click **Details**.
   c. Select Configuration Properties.

- If applying changes to a device:
   a. Click the device in the device list.
   b. Select Configuration Properties.

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

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

Table 39.2. JBoss Configuration Properties

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.

Tip

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."

39.4. 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.

Name	Description
JBoss Core	Core information about any JBoss server, including memory usage, threads, and uptime.
JBoss JCA Connection Pool	
JBoss JGroups Channel	
JBoss JMS Cache	
JBoss JMS Destination	
JBoss JMS Topic	
JBoss Message Driven EJB	

Table 39.3. JBoss Templates

5. Click the OK button to save your changes.

39.5. Viewing Raw Data

See the Section 10.5, “Using **JConsole** to Query a JMX Agent” section for more information about how to investigate raw data returned from the application.

39.6. Daemons

Type	**Name**
Performance Collector	**zenjmx**

Table 39.4. Daemons

Chapter 40. Juniper Devices

40.1. About

The JuniperMonitor ZenPack allows system administrators to monitor their Juniper devices.

40.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.JuniperMonitor

Table 40.1. Juniper Prerequisites

40.3. Enable Monitoring

40.3.1. Configuring Juniper Devices to Allow SNMP Queries

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

40.3.2. Configuring Zenoss

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

1. Navigate to the device or device class in the Zenoss interface.
   - If applying changes to a device class:
     a. Select the class in the devices hierarchy.
     b. Click **Details**.
     c. Select Configuration Properties.
   - If applying changes to a device:
     a. Click the device in the device list.
     b. 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.
zSnmpMonitorIgnore	This should be set to `False`
zSnmpPort	The default port is 161.
zSnmpVer	This should be set to `v2c`

Table 40.2. Juniper Configuration Properties

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 placeholders for graphs. After approximately fifteen minutes you should see the graphs start to become populated with information.

40.4. Daemons

Type	**Name**
Modeler	**zenmodeler**
Performance Collector	**zenperfsnmp**

Table 40.3. Daemons

Chapter 41. LDAP Authentication

41.1. About

The LDAPAuthenticator Enterprise ZenPack allows Zenoss to use your existing LDAP authentication infrastructure (such as Active Directory or OpenLDAP) to enable single sign-on to the Zenoss Web interface. For example, you can reuse the user management tools with which you are familiar to enable your Windows users to use their Windows credentials to authenticate to the Zenoss interface. This saves you from having to manually create user accounts and separately maintain passwords.

The benefits of using a service like LDAP to maintain user accounts and privileges include:

- Does not require users to remember yet another password. This decreases support and maintenance requirements.
- Allows 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.

41.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.LDAPAuthenticator

Table 41.1. LDAP Authentication Prerequisites

41.2.1. LDAP Configuration Information

Before configuring LDAP authentication, you must gather the following information from your LDAP or Active Directory administrator. Here is a list of the required information:

- Hostname or IP address of an Active Directory global catalog server. (Active Directory authentication only)
- Hostname or IP address of an LDAP server. (other LDAP server authentication only)
- User's base Distinguished Name (DN). For example, if your domain was ad.zenoss.com, then your user's base DN might be:

  ```
  cn=users,dc=ad,dc=zenoss,dc=com
  ```

- Manager DN. This is the DN (distinguished name) of a user in the domain administrators group. An example that follows the user's base DN is:

  ```
  cn=Administrator,cn=users,dc=ad,dc=zenoss,dc=com
  ```

- Optional: Active Directory groups to map to Zenoss roles. You can choose to control user roles within the Zenoss Web interface using Active Directory groups instead of controlling the roles directly from within Zenoss. If you do choose to do this you should create the following groups within Active Directory.
  - Zenoss Managers
  - Zenoss Users

Note

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

41.3. Limitations

You cannot use LDAP SSL on CentOS4 or the Zenoss Appliance.

41.4. Authenticating with Microsoft Active Directory

41.4.1. Adding the Authentication Plugin

To add the plugin, you must access the ZMI (Zope Management Interface). This allows raw access to the Zope application server and its configured objects. These steps show how to add the ActiveDirectory Multi Plugin with its default settings.

1. Browse to this URL:

   http://*YourZenossInstallation*:8080/zport/acl_users/manage

2. Choose the ActiveDirectory Multi Plugin plugin, and then click Add.
3. Complete the form with your credentials and paths:

Name	Description
ID	Enter `adPlugin`.
Title	Enter a title, or leave blank.
LDAP Server[:port]	Specify the address of the global catalog server from the pre-requisites section. It should either be the resolvable hostname or IP address of the global catalog server followed by :3268 Example: ad1.zenoss.com:3268 If using SSL, the name must be specified.
Read-only	Select this option.
Users Base DN	Use the value obtained from your AD administrator.
Group storage	Groups not stored on LDAP server.
Groups Base DN	Use the value obtained from your AD administrator.
Manager DN	Use the value obtained from your AD administrator.
Password	Use the value obtained from your AD administrator.

Table 41.2. Active Directory Multi Plugin Configuration

4. Click Add to save your changes.

41.4.2. Configuring Plugin Settings

The default plugin settings need some customizations.

1. Browse to this URL:

   http://*yourzenossinstallation*:8080/zport/acl_users/adPlugin/manage

2. Check the following boxes:
   - Authentication

- Properties
- User_Enumeration
- Roles ([Select only if a default role other than Anonymous is desired.])
- Role_Enumeration ([Select only if a default role other than Anonymous is desired.])

3. Click Update to save your changes.
4. Click Contents tab.
5. Click acl_users folder.
6. Set the following:

Name	Description
User ID Attribute	`Windows Login Name (sAMAccountName)`
RDN Attribute	`Windows Login Name (sAMAccountName)`

Table 41.3. Active Directory acl_users Folder Customizations

7. Click Apply Changes to save your changes.
8. Click LDAP Schema tab.
9. In the Add LDAP schema item section, set the following:

Name	Description
LDAP Attribute Name	mail
Friendly Name	Email Address
Multi-valued	No
Map to Name	email

Table 41.4. Active Directory Schema Item Configuration

10. Click Apply Changes to save your changes.
11. Click Add to save your changes.

41.4.3. Enabling Group to Role Mapping

You can optionally control your users' roles within Zenoss by using the Active Directory groups. If you choose not to do this, you can control their access by setting their roles within the user management section of the Zenoss Web interface. If you choose to use Active Directory groups, you should use the following steps.

1. Browse to one of the following URLs:
   - For LDAP:

     http://*yourzenossinstallation*:8080/zport/acl_users/manage
   - For Active Directory:

     http://*yourzenossinstallation*:8080/zport/acl_users/adPlugin/manage
2. Put a check in Roles and click Update.
3. Click Properties tab.
4. Change the groupid_attr to: `cn`.
5. Click Save Changes to save your changes.

6. Click Contents tab.
7. Click acl_users folder.
8. Set the following:

Name	Description
Group storage	Groups stored on LDAP server
Group mapping	Manually map LDAP groups to Zope roles

Table 41.5. Active Directory Group to Role Configuration

9. Click Apply Changes to save your changes.
10. Click Groups tab.
11. Scroll to the bottom of the page and in the Add LDAP group to Zope role mapping section:
    a. Choose Zenoss Managers on the left and Manager on the right.
    b. Click Add.
    c. Choose Zenoss Users on the left and ZenUser on the right.
    d. Click Add.
    e. Click Apply Changes to save your changes.

41.4.4. Verifying Connectivity and Credentials Outside of Zenoss

Verify your credential information is valid from the Zenoss server by using the **ldapsearch** command. To install this command, use the following for RPM-based systems:

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

For the appliance, use the command:

```
# conary update openldap-clients
```

as the zenoss user on the Zenoss server:

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

41.5. Authenticating with other LDAP Servers

1. Browse to this URL:

   http://`yourzenossinstallation`:8080/zport/acl_users/manage
2. Choose the LDAP Multi Plugin plugin, and then click Add.
3. Complete the form with your LDAP credentials and paths:

Name	Description
ID	Enter `ldapAuthentication`.
Title	Enter a title or leave blank.
LDAP Server[:port]	Specify the name or IP address of the LDAP server. The default port is 389, and the default port for SSL is 636, so the port doesn't need to be specified if using the defaults. If using SSL, the name *must* be specified.

Name	Description
Default User Roles	Set to `ZenUser`. If this is set as blank, LDAP users will not be able to log in.

Table 41.6. LDAP Multi Plugin Configuration

4. Click Add to save your changes.
5. Click plugins in the list of objects.
6. Click the Authentication Plugins link.
7. Move your ldapAuthentication plugin to the list of active plugins, above the userManager plugin

41.6. Optimizing Authentication with a Cache

Once you have configured third-party authentication, you should enable caching. Without a cache of LDAP responses, your Zenoss server must repeatedly query the configured LDAP server or servers for user, group and authentication information. The following steps describe the process of setting up caching of LDAP responses.

1. Log in to Zenoss as a user with the Manager role.
2. Navigate to `/zport/acl_users/manage` in the Web interface. Do this by replacing the end of your URL within the Zenoss Web interface.
3. Choose `RAM Cache Manager` from the list of options at top-right.

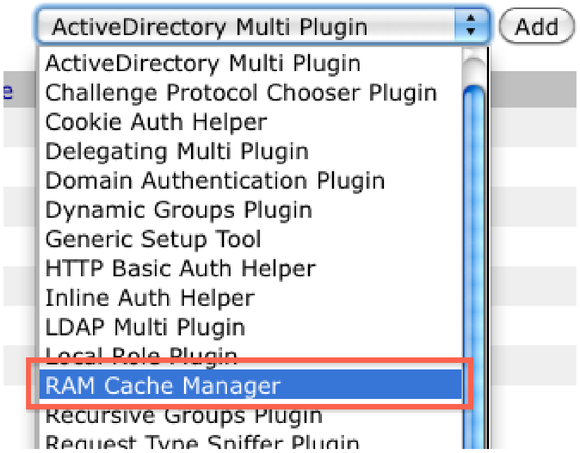


Figure 41.1. Add RAM CAche

4. Set the Id to RAMCache, and then click Add.
5. Click the new RAMCache added to the list to configure it.
   a. Erase AUTHENTICATED_USER from the REQUEST variables field.
   b. Click **Save Changes**.
6. Click acl_users in the breadcrumbs to go back to the acl_users folder.

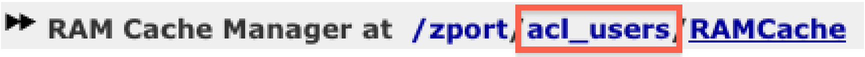


Figure 41.2. acl_users Breadcrumbs

7. Click the Cache tab.

8. Select `RAMCache` from the Cache this object using list, and then click Save Changes.
9. Click the Contents tab.
10. Click the adPlugin or ldapAuthentication plugin, and then click the Cache tab.
11. Select `RAMCache` from the list and click **Save Changes**.

41.7. 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://yourzenoss: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.)

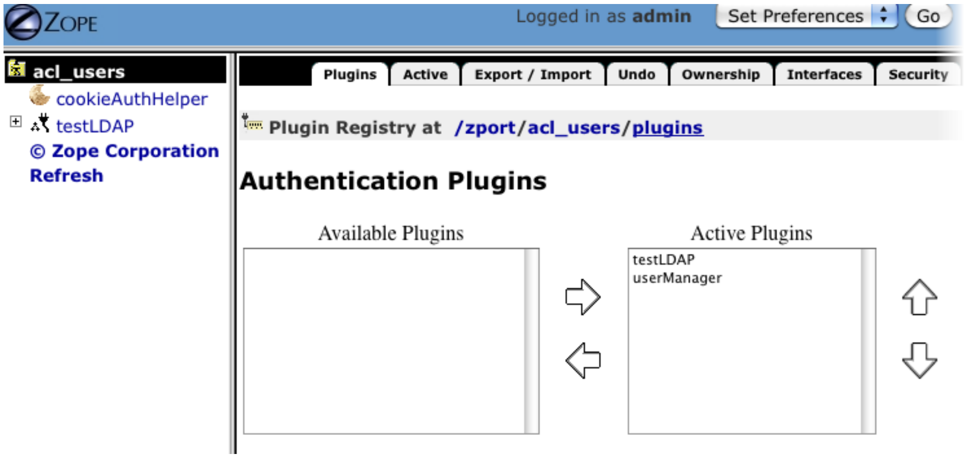


Figure 41.3. Authentication Plugins

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 Zenoss will be valid even when the LDAP server is down.

Chapter 42. Mail Transactions

42.1. About

The ZenMailTx ZenPack allows you to monitor round-trip email delivery.

42.1.1. Events

There are several situations for which ZenMailTx will create 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.

42.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.ZenMailTx

Table 42.1. Mail Transactions Prerequisites

42.3. 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 + 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.

Option	Description
To Address	The e-mail address that will appear in the `From:` field in the generated e-mail.
From Address	The e-mail address where the generated e-mail should be sent.
SMTP Host	The e-mail server where the e-mail should be sent.

Option	Description
POP Host	The POP server from which the test e-mail will be received.

Table 42.2. Mail Transactions Basic Data Source Options

Tip

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.

42.4. Daemons

Type	Name
Performance Collector	**zenmailtx**

Table 42.3. Daemons

Chapter 43. MS Active Directory

43.1. About

The ActiveDirectory ZenPack allows you to monitor Microsoft Active Directory authentication metrics.

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 the Active Directory ZenPack to monitor these 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

43.2. Prerequisites

Prerequisite	**Restriction**
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.ZenWinPerf, ZenPacks.zenoss.ActiveDirectory

Table 43.1. Active Directory Monitoring Prerequisites

43.3. 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 Zenoss Windows service account has access to the Active Directory service.

1. Navigate to the device or device class in the Zenoss interface.

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

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

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

Table 43.2. Active Directory Configuration Properties

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.

43.4. Daemons

Type	Name
Performance Collector	**zenwinperf**

Table 43.3. Daemons

Chapter 44. MS Exchange

44.1. About

The MS Exchange ZenPack is an application monitoring ZenPack that monitors Microsoft Exchange and its related services. The ZenPack enables users to view graphs based on MS Exchange Performance Counters and to monitor processes related to MS Exchange.

44.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.ZenWinPerf, ZenPacks.zenoss.MSExchange

Table 44.1. MS Exchange Prerequisites

44.3. Enable Monitoring

All MS Exchange services must have a device entry under the `/Devices/Server/Windows/MSExchange` device class. In addition, verify that your Zenoss Windows service account has access to the MS Exchange service.

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

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

   Table 44.2. MS Exchange Configuration Properties
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.

44.4. Daemons

Type	Name
Performance Collector	**zenwinperf**

Table 44.3. Daemons

Chapter 45. Microsoft Message Queuing (MSMQ) Monitoring

45.1. About

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.”

The MSMQMonitor ZenPack described in this chapter allows Zenoss to automatically discover queues and monitor how many messages are queued in each.

45.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.5 or higher
Required ZenPacks	ZenPacks.zenoss.MSMQMonitor ZenPacks.zenoss.ZenWinPerf

Table 45.1. MSMQ Monitoring Prerequisites

45.3. Configuration

To monitor the MSMQ queues you must first follow the instructions in the Windows Performance chapter of this guide to setup proper credentials for Zenoss to remotely monitor your Windows server. Once this is done you can take one of the following two approaches to enabled MSMQ queue monitoring.

45.3.1. Automatically Monitor Queues on All Servers

The easiest way to configure Zenoss to monitor your queues is to enable queue discovery for the entire /Server/Windows device class. Within 12 hours Zenoss 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.

45.3.2. Monitor Queues on Specific Servers

If you don't want Zenoss 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.

45.3.3. Fine-Tuning Queue Monitoring

By default Zenoss 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.

45.4. Daemons

Type	Name
Modeler	**zenmodeler**
Performance Collector	**zenwinperf**

Table 45.2. MSMQ Monitoring Daemons

Chapter 46. Microsoft Internet Information Services (IIS)

46.1. About

The IISMonitor ZenPack collects key metrics from Microsoft IIS. The metrics are collected using Windows Perfmon and does not require an agent to be installed on the IIS server.

- 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

46.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.ZenWinPerf, ZenPacks.zenoss.IISMonitor

Table 46.1. MS IIS Prerequisites

46.3. 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 Zenoss 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 Zenoss interface.
   - If applying changes to a device class:
     a. Select the class in the devices hierarchy.
     b. Click **Details**.
     c. Select Configuration Properties.
   - If applying changes to a device:
     a. Click the device in the device list.

   b. Select Configuration Properties.

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

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

Table 46.2. IIS Configuration Properties

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.

46.4. Daemons

Type	Name
Performance Collector	**zenwinperf**

Table 46.3. Daemons

Chapter 47. Microsoft SQL Server

47.1. About

The MSSQLServer ZenPack monitors Microsoft SQL Server and its related services. The ZenPack enables users to view graphs based on Microsoft SQL Server Performance Counters and to monitor processes related to SQL Server.

47.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.ZenWinPerf, ZenPacks.zenoss.MSSQLServer

Table 47.1. MS SQL Server Prerequisites

47.3. 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 Zenoss Windows service account has access to the MS SQL Server service.

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

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

Table 47.2. MS SQL Server Configuration Properties

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.

47.4. Collecting Information from Non-Default Microsoft SQL Server Instances

The default Microsoft SQL Sever instance is SQLServer. The monitoring template delivered with the MSSQLServer ZenPack uses this default instance to gather performance metrics. If you use a non-default SQL Server instance, then Zenoss does not automatically find and gather information about it.

To enable Zenoss 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.

47.5. Daemons

Type	Name
Performance Collector	**zenwinperf**

Table 47.3. Daemons

Chapter 48. Multi-Realm IP Networks

48.1. About

The Multi-Realm IP ZenPack functionality extends core modeling, monitoring, and event management in Zenoss to allow for overlapping IP spaces. With this ZenPack, Zenoss can prefix a realm identifier to the IP addresses on a given network to differentiate these addresses in Zenoss.

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 Zenoss 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.

Note

The Multi-Realm IP ZenPack is available only by separate download from the Zenoss Support site.

After downloading the ZenPack, you must install it manually. In the Zenoss interface, go to Settings > Zenpacks > Install Zenpack.

48.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.DistributedCollector, ZenPacks.zenoss.MultiRealmIP

Table 48.1. Multi-realm Prerequisites

48.3. Example System

The following diagram lays out an example setup. It has a central Zenoss server in the 10.10.10.0/24 network. The network local to the Zenoss server is considered the default network within the system. The default network is treated exactly the same as a Zenoss system without Multi-Realm IP 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 Zenoss 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.

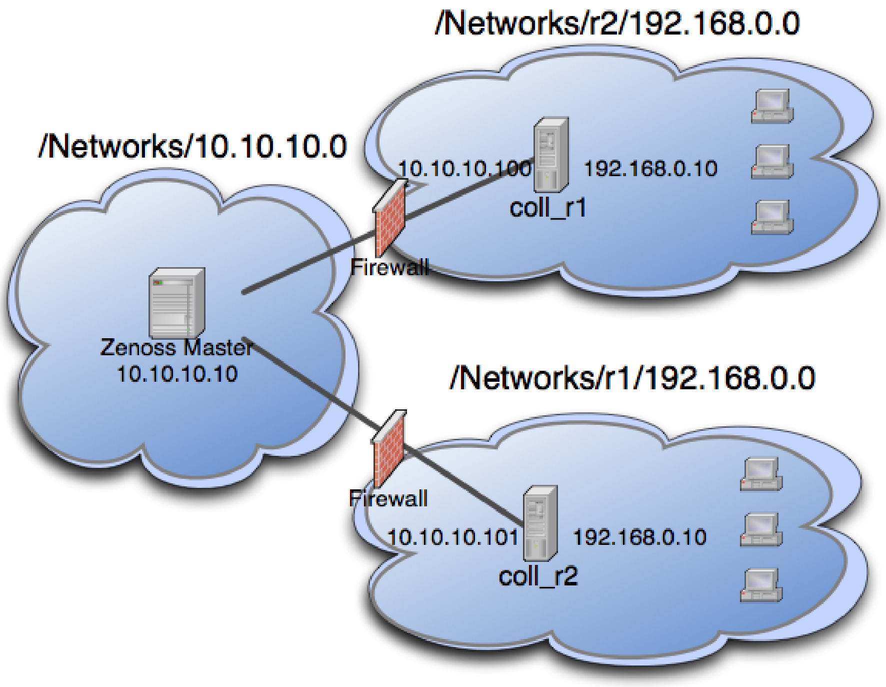


Figure 48.1. Example IP Realm

48.4. System Setup

Set up Zenoss following the example system described above.

Tip

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 Zenoss uses to access MySQL, and the second 'zenoss' is that account's password.

48.4.1. Adding Realms

1. Go to Infrastructure > Networks.
2. From the Add menu, select Add IP Realm. Add the realms `r1` and `r2`.

48.4.2. 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.

48.4.3. 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.

48.5. 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.

Chapter 49. NetApp Filers

49.1. About

NetAppMonitor provides additional modeling and monitoring for NetApp devices. NFS, CIFS and HTTP operations per second are collected, as well as file system and snapshot utilization information. Hardware model and operating system revision asset information is modeled.

The NetApp ZenPack uses reports (IIRC) provided by the StorageBase ZenPack.

Asset information:

- Hardware Model
- Operating System Revision

Device metrics:

- Network bits/sec: Send and Received
- Operations/sec: NFS, CIFS and HTTP

File system metrics:

- File system utilization (90% threshold)
- Snapshot utilization (120% threshold)

NetApp uses SSH to model NFS clients of file systems. It uses SNMP to model:

- Disks, storage enclosures, RAID groups, Plexes, Aggregates, Volumes, LUNs and QTrees
- LUN clients
- Licenses

Note

Sizes reported by the NetAppMonitor ZenPack are approximate, as values for many objects (Aggregate, Volume, Plex, and RAID group) are not exposed by the NetApp MIB.

It uses SNMP to monitor:

- iSCSI, Fibre Channel, and per LUN throughput
- Disk inventory (active, spare, pre-failed, or failed)
- Disk maintenance activity (scrubbing, reconstructing, parity reconstructing, verifying parity)
- NFS v3 statistics
- NFS cache statistics
- CIFS statistics

49.1.1. Performance Graphs

Performance graphs provided with this ZenPack include:

- NFSv3 Operations
- Fibre Channel Traffic

- iSCSI Traffic
- NFS Caching Statistics
- Disk Inventory
- Disk Maintenance

49.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 3.x or higher
Required ZenPacks	ZenPacks.zenoss.NetAppMonitor ZenPacks.zenoss.StorageBase

Table 49.1. NetApp Prerequisites

49.3. Enable Monitoring

49.3.1. Configuring NetApp Devices to Allow SNMP Queries

Configure the NetApp devices to allow SNMP queries from the Zenoss server, and send SNMP v1 or SNMP v2 traps to the Zenoss server.

49.3.2. Configuring Zenoss

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

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

Name	Description
zSnmpCommunity	Consult with your storage administrators to determine the SNMP community permitted.
zSnmpPort	The default port is 161.
zSnmpVer	Set to `v2c`.

Table 49.2. NetApp Configuration Properties

3. Click Save to save your changes. You will now be able to start collecting the NetApp 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.

49.4. Using SSH to Model NFS Clients

To use SSH to model NFS clients, you must:

1. Allow SSH logins to the NetApp server.
2. Set the configuration properties zCommandPassword and zCommandUser.
3. Remodel the device.

49.5. Forwarding syslog Events from NetApp

To forward syslog events from NetApp:

1. From the NetApp interface, click the Filer menu.
2. Click the Configure Syslog menu item.
3. Click the New Action button.
4. Add the following, *separating each field with a tab*.

   ```
   *.* @yourzenossserver
   ```
5. Click **OK**.

You can test the configuration by logging in to the command line on the NetApp server, and then entering the following command:

```
logger Hello World
```

This should result in an event with the subject "Hello World" appearing in the Zenoss event console. To restart the daemon, enter:

```
syslog reset_syslog
```

49.6. Daemons

Type	Name
Modeler	**zenmodeler**
Performance Collector	**zenperfsnmp**
Performance Collector	**zencommand**

Table 49.3. Daemons

Chapter 50. NetScreen Devices

50.1. About

NetScreenMonitor allows you to monitor NetScreen devices.

50.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.NetScreenMonitor

Table 50.1. NetScreen Prerequisites

50.3. Enable Monitoring

50.3.1. Configuring NetScreen Devices to Allow SNMP Queries

Configure the NetScreen device to allow SNMP queries from the Zenoss server, and send SNMP v1 or SNMP v2 traps to the Zenoss server.

50.3.2. Configuring Zenoss

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

1. Navigate to the device or device class in the Zenoss interface.
   - If applying changes to a device class:
     a. Select the class in the devices hierarchy.
     b. Click **Details**.
     c. Select Configuration Properties.
   - If applying changes to a device:
     a. Click the device in the device list.
     b. 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.
zSnmpMonitorIgnore	This should be set to `False`
zSnmpPort	The default port is 161.
zSnmpVer	This should be set to `v2c`

Table 50.2. NetScreen Configuration Properties

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.

50.4. Daemons

Type	Name
Modeler	**zenmodeler**
Performance Collector	**zenperfsnmp**

Table 50.3. Daemons

Chapter 51. Nortel Devices

51.1. About

The NortelMonitor ZenPack allows you to monitor Nortel devices.

51.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.NortelMonitor

Table 51.1. Nortel Prerequisites

51.3. Enable Monitoring

51.3.1. Configuring Nortel Devices to Allow SNMP Queries

Configure the Nortel device to allow SNMP queries from the Zenoss server, and send SNMP v1 or SNMP v2 traps to the Zenoss server.

51.3.2. Configuring Zenoss

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

1. Navigate to the device or device class in the Zenoss interface.
   - If applying changes to a device class:
     a. Select the class in the devices hierarchy.
     b. Click **Details**.
     c. Select Configuration Properties.
   - If applying changes to a device:
     a. Click the device in the device list.
     b. 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.
zSnmpMonitorIgnore	This should be set to `False`
zSnmpPort	The default port is 161.
zSnmpVer	This should be set to `v2c`

   Table 51.2. Nortel Configuration Properties

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.

51.4. Daemons

Type	Name
Modeler	**zenmodeler**
Performance Collector	**zenperfsnmp**

Table 51.3. Daemons

Chapter 52. Oracle

52.1. About

The Oracle Monitoring ZenPack (DatabaseMonitor) monitors an Oracle database server. The ZenPack enables users to view graphs based on interface from Oracle performance tables.

52.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.DatabaseMonitor

Table 52.1. Oracle Prerequisites

Note

The Oracle ZenPack (ZenPacks.zenoss.DatabaseMonitor) is not available at the Enterprise Download site by default for legal reasons. It is also not included in the Enterprise ZenPacks RPM file.

Oracle requires each user to complete a license agreement prior to receiving this ZenPack. Upon completion, Zenoss Support will enable the ZenPack. You will be notified via a new case in the Zenoss Support Portal when the ZenPack is available. This ZenPack will be located in the `zenpacks` directory at the Enterprise Download site as both a 32-bit and a 64-bit version.

After downloading the ZenPack, you must install it manually. In the Zenoss interface, go to Advanced > Settings, select ZenPacks in the left panel, and then select Install ZenPack from the Action menu.

52.3. Enable Monitoring

52.3.1. Authorize Oracle Performance Data Access

The default Oracle monitoring template queries the `v$statname` and `v$sysstat` views for performance metrics. You must get a login to the Oracle instance with read privileges to these tables at the minimum. You must also provide read privileges any other custom tables or views you plan to monitor.

52.3.2. Configure Zenoss

Oracle monitoring can be applied to any device in the system by binding the Oracle template and configuring a few properties. The following steps illustrate how you would add Oracle monitoring to a Windows server called `oraprod1.example.com`.

1. Select the `oraprod1.example.com` device in the device list.
2. In the left panel, select Configuration Properties.
3. Set the following Oracle-related properties.
   - zOracleConnectString: Optionally used instead of the following separate options.
   - zOracleInstance: Oracle SID
   - zOraclePassword: Password for the Oracle account
   - zOraclePort: Port number for the Oracle instance
   - zOracleUser: Username for the Oracle account

4. Click Save.
5. Select Bind Templates from the Action menu.
6. Move the Oracle template from the list of Available templates to the Selected area.
7. Click Save.

You will now be able to find the following additional graphs on the device. It may take up to fifteen minutes to start displaying values. You can check the device's event console for any errors related to the Oracle collection.

52.4. Monitor Additional SIDs

To monitor performance data from an additional SID on the same device you must make a copy of the default Oracle template and adjust its instance property.

1. Navigate to Advanced -> Monitoring Templates.
2. Highlight the /Devices under the Oracle template.
3. Choose Override Template from the action menu.
4. Choose your Oracle server from the list.
5. Click Submit.
6. Highlight the new Oracle template for your Oracle server.
7. Choose View and Edit Details from the gear menu.
8. Change the template name from Oracle to Oracle??? where ??? is the SID.
9. Click Submit.
10. Highlight the Oracle template for your Oracle server.

52.5. Monitoring Other Tables or Views

The Oracle data source also allows monitoring of other data contained within the database. You will need to build a query that returns a table in the following format.

Data Point Name	Numeric Value
firstValue	123
secondValue	45.6

Table 52.2. Example Query Results

Once you have a result set conforming to this name, value column specification you can add a new Oracle data source to a new or existing template using the following steps.

1. Optionally create a new monitoring template for the data source.
2. Edit the monitoring template.
3. Add a new Oracle data source to the template.
4. Fill out all of the data source fields as required to make the query.
5. Add one data point to the new data source for each row.

Note

The data point name must match the value in the first column of the result set exactly. For the example result set shown above you would create a data point named `firstValue` and another named `secondValue`.

52.6. Daemons

Type	**Name**
Performance Collector	**zencommand**

Table 52.3. Daemons

Chapter 53. Predictive Thresholding

53.1. About

The 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 this prediction.

For more information on RRD and Holt-Winters, see the **rrdcreate** command for more information.

Warning

Zenoss 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.

53.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.ZenHoltWinters

Table 53.1. Trap Forwarding Prerequisites

53.3. 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.

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.
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.

Table 53.2. Predictive Threshold Data Source Threshold Options

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.

Chapter 54. RANCID Integration

54.1. About

The RANCIDIntegrator ZenPack allows integration between the popular RANCID configuration management tool and Zenoss. The integration points between the tools are:

- Zenoss 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.
- Zenoss 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 Zenoss. 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 Zenoss master and as such works only with managed resources on the master.

54.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.RANCIDIntegrator

Table 54.1. RANCID Prerequisites

54.3. Enable Integration

54.3.1. Configure Cisco Devices to Send Traps

To implement this feature you must configure your Cisco devices to send their SNMP traps to the Zenoss server.

Link from Cisco device status pages to the most recent configuration stored in your CVS repository via viewvc.

54.3.2. Configure RANCID Update Information in Zenoss

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.

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

Name	Description
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`

Table 54.2. RANCID Configuration Properties

4. Click Save to save your changes.

Chapter 55. Remedy Ticket Creation

55.1. About

The RemedyIntegrator ZenPack provides a way for Zenoss to automatically open tickets in your Remedy system when specific events occur. The cases are opened by Zenoss sending a specially-formatted email to the Remedy service's email receiver.

55.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.RemedyIntegrator

Table 55.1. Remedy Ticket Creation Prerequisites

55.3. Enable Ticket Creation

Have your Remedy administrator create the template email that is needed to have events enter into the appropriate workflow.

1. Select Advanced > Settings from the navigation bar.
2. Select Users from the left panel.
3. Click on the `Remedy` user.
4. Change the email address to the email address from which your Remedy service receives email.
5. Select Alerting Rules in the left panel.
6. Click on the Open Ticket alerting rule to edit it.
7. Set the Enabled field to `True` and adjust the event filter to your requirements.
8. Select Message in the left panel and modify the Zenoss e-mail message with the necessary information from the template e-mail. Lines that will require modification for all sites are:
    - Server: remedy.yourdomain.com
    - Login: remedyUsername
    - Password: remedyPassword
9. Click Save to save your changes.

55.4. Send Test Tickets

To create test events that will match your rule and create tickets in Remedy, select Events from the navigation bar, and then click ➕ (Add Event).

55.5. Daemons

Type	Name
Event Forwarder	**zenactions**

Table 55.2. Daemons

Chapter 56. SNMP Trap Forwarding

56.1. About

Zenoss can be configured to forward events matching specified criteria to other SNMP trap receivers. You may want to do this if you have another system that would benefit from the event information that Zenoss collects.

Note

This ZenPack is available only by separate download from the Zenoss Support site. After downloading the ZenPack, you must install it manually:

1. In the Zenoss interface, go to Advanced > Settings.
2. In the left panel, select ZenPacks.
3. Select Install ZenPack from the Action menu.

56.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.TrapForwarder

Table 56.1. Trap Forwarding Prerequisites

56.3. Enable Event Forwarding

56.3.1. Import Zenoss MIB onto the Remote Receiver

The MIB file `ZENOSS-MIB.txt` is found at the base directory within the TrapForwarder distribution. Import this MIB into the event management system that you plan to forward to events to so that the SNMP traps that Zenoss will generate can be properly interpreted. Consult the documentation for the remote SNMP manager for instructions.

56.3.2. Configure Zenoss to Send Events as Traps

1. From the navigation bar, select Advanced > Settings.
2. Select Daemons in the left panel.
3. Locate the trapforwarder daemon in the list, and then click edit config in that row.
4. Specify the following properties that are expected by your remote SNMP trap receiver.

Name	Description
community	SNMP community name sent in each trap
trapsink	hostname or IP address of the remote SNMP trap receiver
Required ZenPacks	ZenPacks.zenoss.TrapForwarder

Table 56.2. trapforwarder Configuration File Options

5. Click **Save** to save your changes.
6. Select Daemons in the left panel and locate the trapforwarder daemon. Click **Restart** in that row.
7. From the navigation bar, select Events > Event Manager.

8. Select Commands in the left panel.
9. Choose the events you want to forward, based on the example already setup called SNMP Trap. Click to edit it.

   The example is set up to forward all new events from production devices with a severity of warning or greater. You may want to limit this further.
10. Once configuration of the rule is complete, select a value of True for the Enabled field.
11. Click Save to save your changes.

56.4. Send Test Events

To create test events that will match your rule, navigate to Events from the navigation bar and then add an event.

56.5. Daemons

Type	**Name**
Event Forwarder	**trapforwarder**

Table 56.3. Daemons

Chapter 57. Solaris

57.1. About

The SolarisMonitor ZenPack enables Zenoss to use Secure Shell (SSH) to monitor Solaris hosts. Zenoss models and monitors devices placed in the `/Server/SSH/Solaris` device class by running commands and parsing the output. Parsing of command output is performed on the Zenoss server (if using a local collector) or on a distributed collector. The account used to monitor the device does not require root access or special privileges.

The SolarisMonitor 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)
- Pkginfo information (such as installed software)

57.2. Prerequisites

Prerequisite	**Restriction**
Zenoss Version	Zenoss Version 2.5 or higher
Required ZenPacks	ZenPacks.zenoss.SolarisMonitor
Solaris releases supported	OpenSolaris 5.11, Solaris 9 and 10

Table 57.1. Solaris Prerequisites

Note

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

57.3. Limitations

The SolarisMonitor ZenPack does not support monitoring in Solaris Zones or systems containing Solaris Zones. (Implemented with Solaris 10, Solaris Zones act as isolated virtual servers within a single operating system instance.)

57.4. Set Solaris Server Monitoring Credentials

All Solaris servers must have a device entry in an organizer below the `/Devices/Server/SSH/Solaris` 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 Zenoss interface.
    - If applying changes to a device class:
        a. Select the class in the devices hierarchy.
        b. Click **Details**.

   c. Select Configuration Properties.

- If applying changes to a device:
   a. Click the device in the device list.
   b. Select Configuration Properties.

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

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

Table 57.2. Solaris Configuration Properties

3. Click Save to save your changes.

57.5. Enable Monitoring

These steps assume that credentials have been set.

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

Name	Description
Device Name	Solaris host to model
Device Class Path	`/Server/SSH/Solaris`
Discovery Protocol	Set this to `auto` unless adding a device with a username and password different than found in the device class. If you set this to `none`, then you must add the credentials (see Section 57.4, “Set Solaris Server Monitoring Credentials”), and then manually model the device.

Table 57.3. Adding Solaris Device Information

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

57.6. 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. 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 Zenoss interface.
   - If applying changes to a device class:

a. Select the class in the devices hierarchy.

b. Click **Details**.

c. Select Configuration Properties.

- If applying changes to a device:

a. Click the device in the device list.

b. Select Configuration Properties.

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

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

Table 57.4. Concurrent SSH Configuration Properties

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

57.7. 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 usually 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 Zenoss interface.
   - If applying changes to a device class:

a. Select the class in the devices hierarchy.

b. Click **Details**.

c. Select Configuration Properties.

- If applying changes to a device:

a. Click the device in the device list.

b. Select Configuration Properties.

2. Apply an appropriate value for the command timeout.

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

Table 57.5. SSH Timeout Configuration Properties

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

57.8. Daemons

Type	Name
Modeler	**zenmodeler**

Type	Name
Performance Collector	**zencommand**

Table 57.6. Daemons

Chapter 58. Splunk Monitoring

58.1. About

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/.

The Splunk 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.

58.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.4 or higher
Required ZenPacks	ZenPacks.zenoss.Splunk
Third Party Software	Splunk Version 3 or 4

Table 58.1. Splunk Monitoring Prerequisites

58.3. Splunk Data Source Type

The Splunk ZenPack adds the new Splunk data source type to your Zenoss system. 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 Zenoss 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.

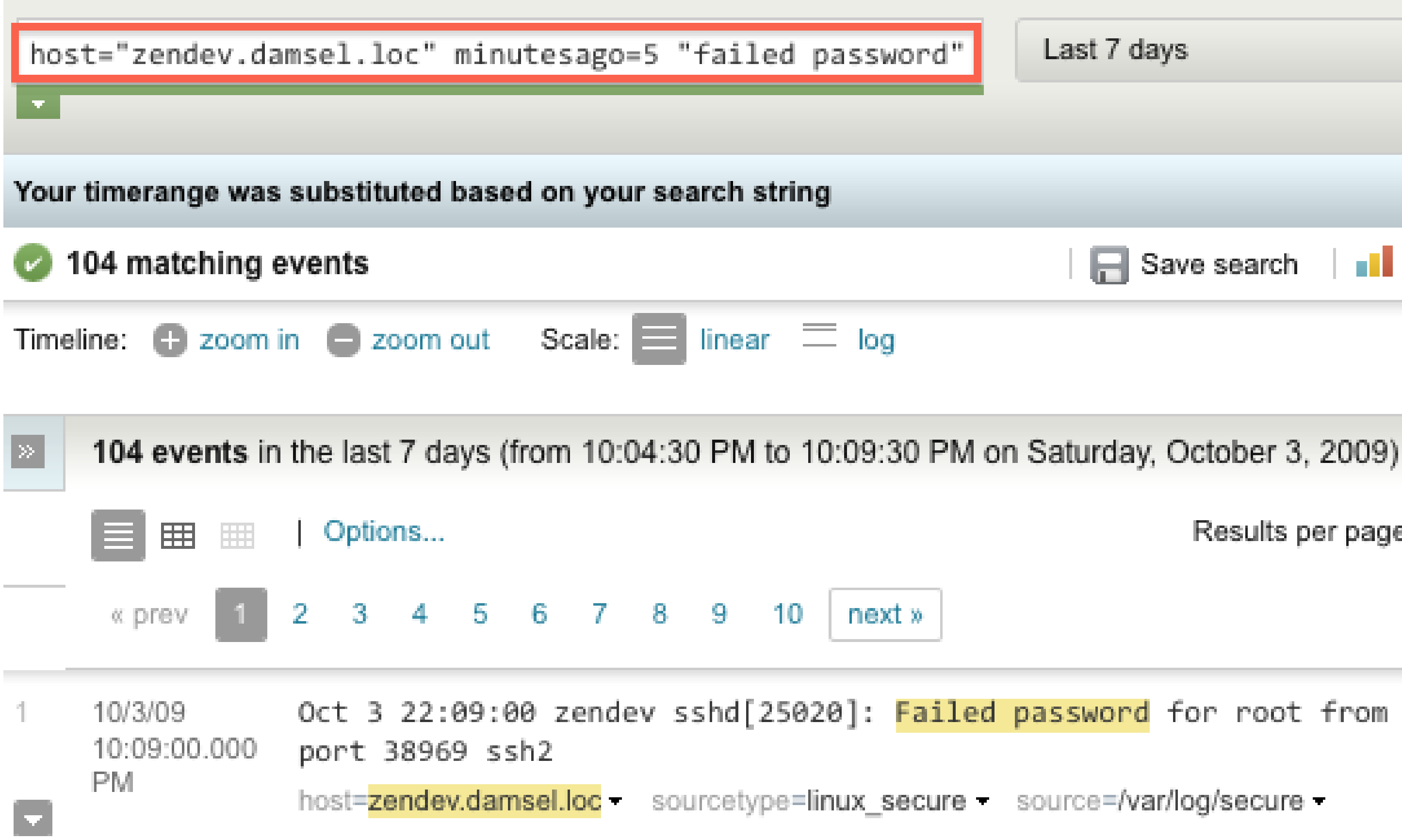
58.4. Monitoring Splunk Searches

58.4.1. 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".



```
host="zendev.damsel.loc" minutesago=5 "failed password"
```

Your timerange was substituted based on your search string

104 matching events

Save search
Timeline: zoom in zoom out Scale: linear log

104 events in the last 7 days (from 10:04:30 PM to 10:09:30 PM on Saturday, October 3, 2009)

Options...
Results per page
« prev 1 2 3 4 5 6 7 8 9 10 next »

1 10/3/09 10:09:00.000 PM

```
Oct 3 22:09:00 zendev sshd[25020]: Failed password for root from
port 38969 ssh2
```

host=zendev.damsel.loc sourcetype=linux_secure source=/var/log/secure

Note

Using a time specifier such as minutesago=5 within your search can be a useful trick when it comes to monitoring searches from Zenoss. We will have Zenoss automatically replace zendev.damsel.loc with the appropriate hostname using a ${here/id} TALES expression.

2. Create a Zenoss monitoring template for monitoring this Splunk search.

   a. From Advanced > Monitoring Templates, click + 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.

   i. In the Data Sources area, click + to add a data source.

   ii. In the Add Data Source dialog, set the Name to failedPassword and the Type to `Splunk`, and then click OK.

   iii. 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"

   iv. 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.

   v. Add a threshold of how many failed passwords constitutes an attack.

   A. In the Thresholds area, click + 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.

   vi. Add a graph to visualize failed passwords per 5 minutes.

   A. In the Graph Definitions area, click + 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.

vii. 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**.

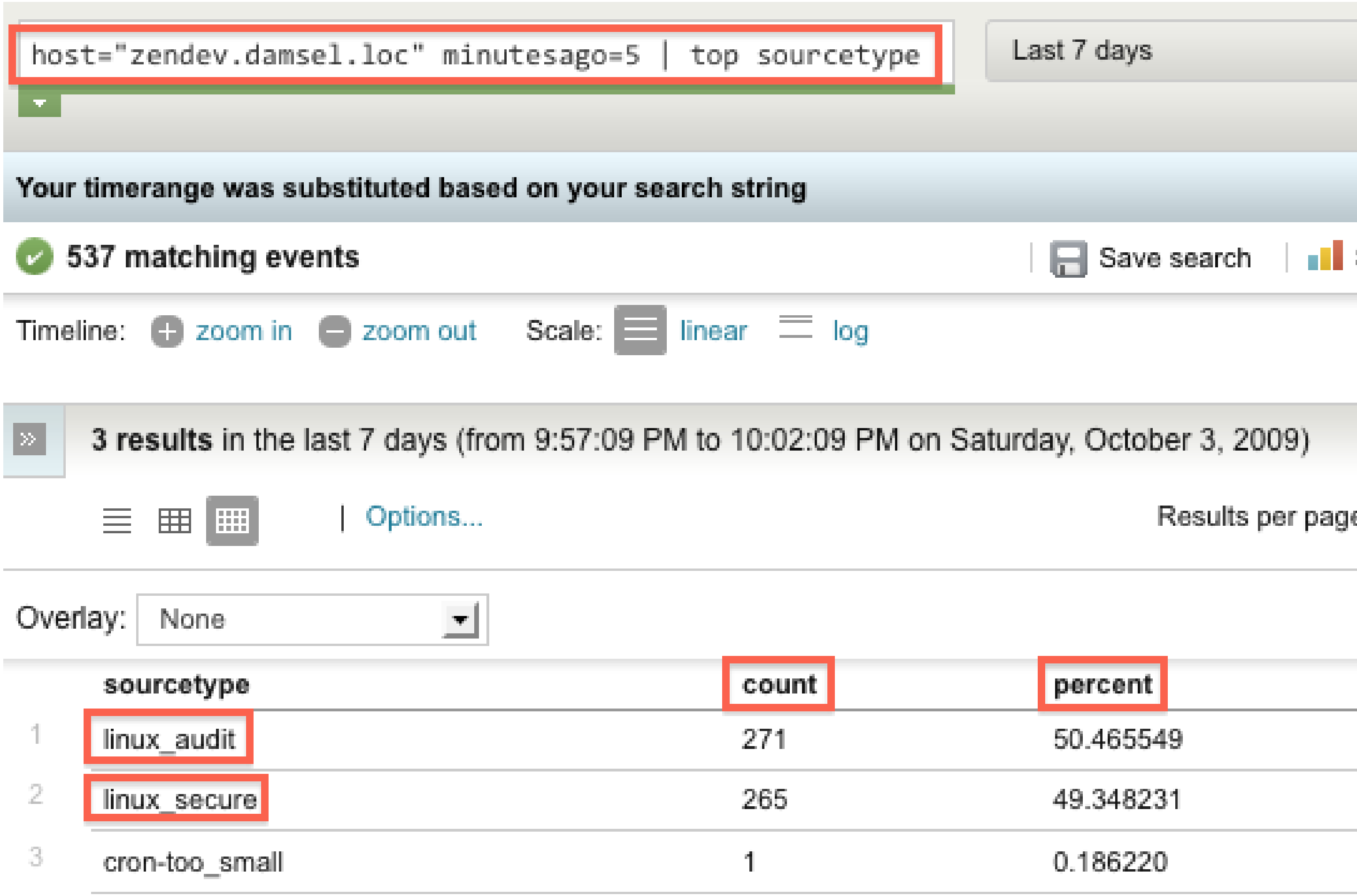
Now you will have a Failed Passwords graph on all of your Linux servers that visualizes how many failed password attempts have occurred over the last 5 minutes. You will also get a warning severity event anytime more than 10 failed password attempts are made within a 5 minute period.

58.4.2. 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.



```
host="zendev.damsel.loc" minutesago=5 | top sourcetype
```

Your timerange was substituted based on your search string

537 matching events

3 results in the last 7 days (from 9:57:09 PM to 10:02:09 PM on Saturday, October 3, 2009)

	sourcetype	count	percent
1	linux_audit	271	50.465549
2	linux_secure	265	49.348231
3	cron-too_small	1	0.186220

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 Zenoss 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 Zenoss 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.

58.5. Daemons

Type	Name
Performance Collector	**zencommand**

Table 58.2. Splunk Monitoring Daemons

Chapter 59. SQL Transactions

59.1. About

The ZenSQLTx ZenPack allows you to test the availability and performance of MySQL, Sybase and Microsoft SQL servers. It provides a SQL data source where user-defined SQL queries can be executed against a database.

59.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.ZenSQLTx

Table 59.1. SQL Transaction Prerequisites

59.3. 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 ➕.
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.

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).

Option	Description
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.

Table 59.2. MS SQL Server Transactions Data Source Options

10. Click Save to save your changes.

    Zenoss 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.

See the Administration Guide for more information about setting up thresholds and graphs. To create data points that store the results of queries, see the section titled "Data Points."

59.4. 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 +.
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.

Option	Description
Database Type	Enter `Sybase`
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.

Option	Description
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.

Table 59.3. MySQL Server Transactions Data Source Options

10. Click on the Save button to save your changes.

    Zenoss 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.

See the Administration Guide for more information about setting up thresholds and graphs. To create data points that store the results of queries, see the section titled "Data Points."

59.5. 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 +.
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.

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.
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:

Option	Description
	http://dev.mysql.com/doc/refman/5.0/en/sql-syntax.html

Table 59.4. MySQL Server Transactions Data Source Options

10. Click on the Save button to save your changes.

    Zenoss 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.

See *Zenoss Administration* for more information about setting up thresholds and graphs. To create data points that store the results of queries, see the section titled "Data Points."

59.6. 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.

59.7. 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 Zenoss 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.

Name	Database Type
`mssql`	MS SQL Server
`sybase`	Sybase
`mysql`	MySQL Server

Table 59.5. zensql.py Database Types

59.8. Daemons

Type	Name
Performance Collector	**zencommand**

Table 59.6. Daemons

Chapter 60. Storage Base

60.1. About

The StorageBase ZenPack contains base classes, and reports for ZenPacks that use those base classes.

The ZenPack includes these reports:

- **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.

60.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 3.0 or higher
Required ZenPacks	ZenPacks.zenoss.StorageBase, ZenPacks.zenoss.DynamicView

Table 60.1. Prerequisites

Chapter 61. Sugar CRM

61.1. About

SugarCRMMonitor is a ZenPack that allows System Administrators to monitor their Sugar CRM services.

61.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.SugarCRMMonitor

Table 61.1. Sugar CRM Prerequisites

61.3. Enable Monitoring

61.3.1. Configuring Zenoss

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 Zenoss interface.
   - If applying changes to a device class:
     a. Select the class in the devices hierarchy.
     b. Click **Details**.
     c. Select Configuration Properties.
   - If applying changes to a device:
     a. Click the device in the device list.
     b. Select Configuration Properties.
2. Edit the appropriate configuration properties for the device or devices.

Name	Description
zSugarCRMBase	
zSugarCRMPassword	Password for the `zSugarCRMUsername` user.
zSugarCRMTestAccount	
zSugarCRMUsername	Username allowed to log into the Sugar CRM server.

   Table 61.2. SugarCRM Configuration Properties
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.

61.4. Daemons

Type	Name
Performance Collector	**zencommand**

Table 61.3. Daemons

Chapter 62. VMware ESX

62.1. About

The VMwareESXMonitor ZenPack allows you to monitor VMware ESX hosts and their guests. This ZenPack:

- Extends ZenModeler to discover guests running on the ESX host.
- Provides screens and templates for collecting and displaying resources allocated to the guests.

This ZenPack requires the ZenossVirtualHostMonitor ZenPack to be installed as a prerequisite.

62.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.VMwareESXMonitor, ZenPacks.zenoss.ZenossVirtualHostMonitor

Table 62.1. Prerequisites

62.3. 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 Zenoss 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 Zenoss, a link is provided to take you directly to that device from the Guest list.

62.4. 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 Zenoss 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
   ```

62.5. Daemons

Type	Name
Modeler	**zenmodeler**
Performance Collector	**zenperfsnmp**

Table 62.2. Daemons

Chapter 63. VMware Virtual Hosts

63.1. About

With Zenoss, you can collect information to monitor your VMware infrastructure. By entering a single set of connection parameters, you allow Zenoss to:

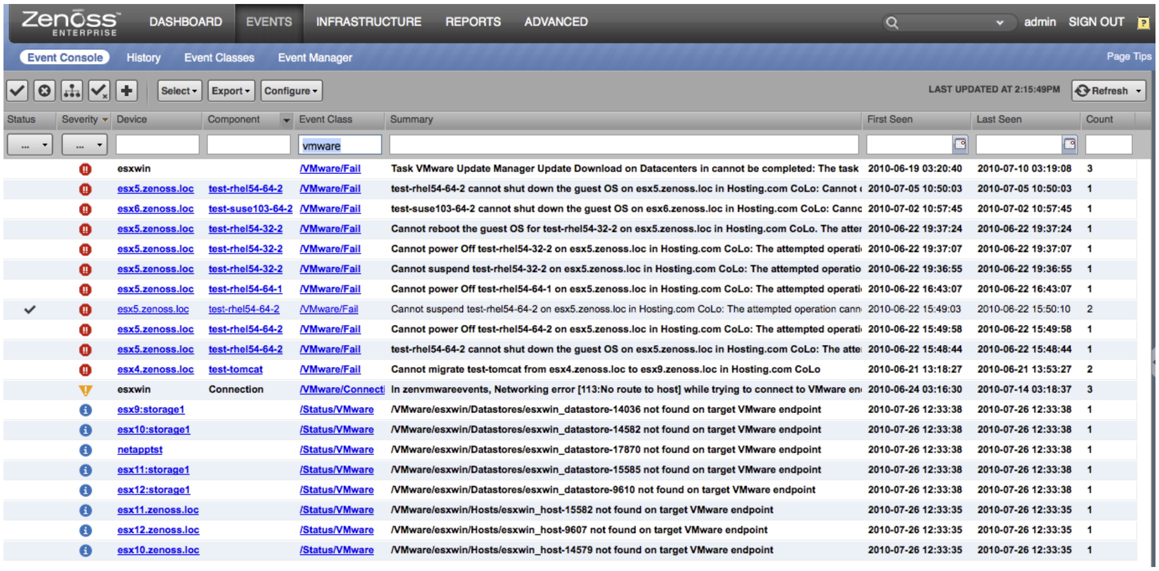
- Obtain the names and properties of various entities in your VMware infrastructure
- Monitor metrics collected by VMware
- Retrieve VMware events

Zenoss 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 Introduction to VMware Infrastructure

63.1.1. VMware Events

VMware records a wide range of events that are available through the VI SDK. Zenoss extracts these events and makes them available in the event console.



Status	Severity	Device	Component	Event Class	Summary	First Seen	Last Seen	Count
...	...			vmware				
	‼	esxwin		/VMware/Fail	Task VMware Update Manager Update Download on Datacenters in cannot be completed: The task	2010-06-19 03:20:40	2010-07-10 03:19:08	3
	‼	esx5.zenoss.loc	test-rhel54-64-2	/VMware/Fail	test-rhel54-64-2 cannot shut down the guest OS on esx5.zenoss.loc in Hosting.com CoLo: Cannot	2010-07-05 10:50:03	2010-07-05 10:50:03	1
	‼	esx6.zenoss.loc	test-suse103-64-2	/VMware/Fail	test-suse103-64-2 cannot shut down the guest OS on esx6.zenoss.loc in Hosting.com CoLo: Cannc	2010-07-02 10:57:45	2010-07-02 10:57:45	1
	‼	esx5.zenoss.loc	test-rhel54-32-2	/VMware/Fail	Cannot reboot the guest OS for test-rhel54-32-2 on esx5.zenoss.loc in Hosting.com CoLo. The atter	2010-06-22 19:37:24	2010-06-22 19:37:24	1
	‼	esx5.zenoss.loc	test-rhel54-32-2	/VMware/Fail	Cannot power Off test-rhel54-32-2 on esx5.zenoss.loc in Hosting.com CoLo: The attempted operati	2010-06-22 19:37:07	2010-06-22 19:37:07	1
	‼	esx5.zenoss.loc	test-rhel54-32-2	/VMware/Fail	Cannot suspend test-rhel54-32-2 on esx5.zenoss.loc in Hosting.com CoLo: The attempted operatio	2010-06-22 19:36:55	2010-06-22 19:36:55	1
	‼	esx5.zenoss.loc	test-rhel54-64-1	/VMware/Fail	Cannot power Off test-rhel54-64-1 on esx5.zenoss.loc in Hosting.com CoLo: The attempted operati	2010-06-22 16:43:07	2010-06-22 16:43:07	1
✓	‼	esx5.zenoss.loc	test-rhel54-64-2	/VMware/Fail	Cannot suspend test-rhel54-64-2 on esx5.zenoss.loc in Hosting.com CoLo: The attempted operation cann	2010-06-22 15:49:03	2010-06-22 15:50:10	2
	‼	esx5.zenoss.loc	test-rhel54-64-2	/VMware/Fail	Cannot power Off test-rhel54-64-2 on esx5.zenoss.loc in Hosting.com CoLo: The attempted operati	2010-06-22 15:49:58	2010-06-22 15:49:58	1
	‼	esx5.zenoss.loc	test-rhel54-64-2	/VMware/Fail	test-rhel54-64-2 cannot shut down the guest OS on esx5.zenoss.loc in Hosting.com CoLo: The atte	2010-06-22 15:48:44	2010-06-22 15:48:44	1
	‼	esx4.zenoss.loc	test-tomcat	/VMware/Fail	Cannot migrate test-tomcat from esx4.zenoss.loc to esx9.zenoss.loc in Hosting.com CoLo	2010-06-21 13:18:27	2010-06-21 13:53:27	2
	⚠	esxwin	Connection	/VMware/Connecti	In zenvmwareevents, Networking error [113:No route to host] while trying to connect to VMware en	2010-06-24 03:16:30	2010-07-14 03:18:37	3
	ℹ	esx9:storage1		/Status/VMware	/VMware/esxwin/Datastores/esxwin_datastore-14036 not found on target VMware endpoint	2010-07-26 12:33:38	2010-07-26 12:33:38	1
	ℹ	esx10:storage1		/Status/VMware	/VMware/esxwin/Datastores/esxwin_datastore-14582 not found on target VMware endpoint	2010-07-26 12:33:38	2010-07-26 12:33:38	1
	ℹ	netapptst		/Status/VMware	/VMware/esxwin/Datastores/esxwin_datastore-17870 not found on target VMware endpoint	2010-07-26 12:33:38	2010-07-26 12:33:38	1
	ℹ	esx11:storage1		/Status/VMware	/VMware/esxwin/Datastores/esxwin_datastore-15585 not found on target VMware endpoint	2010-07-26 12:33:38	2010-07-26 12:33:38	1
	ℹ	esx12:storage1		/Status/VMware	/VMware/esxwin/Datastores/esxwin_datastore-9610 not found on target VMware endpoint	2010-07-26 12:33:38	2010-07-26 12:33:38	1
	ℹ	esx11.zenoss.loc		/Status/VMware	/VMware/esxwin/Hosts/esxwin_host-15582 not found on target VMware endpoint	2010-07-26 12:33:35	2010-07-26 12:33:35	1
	ℹ	esx12.zenoss.loc		/Status/VMware	/VMware/esxwin/Hosts/esxwin_host-9607 not found on target VMware endpoint	2010-07-26 12:33:35	2010-07-26 12:33:35	1
	ℹ	esx10.zenoss.loc		/Status/VMware	/VMware/esxwin/Hosts/esxwin_host-14579 not found on target VMware endpoint	2010-07-26 12:33:35	2010-07-26 12:33:35	1

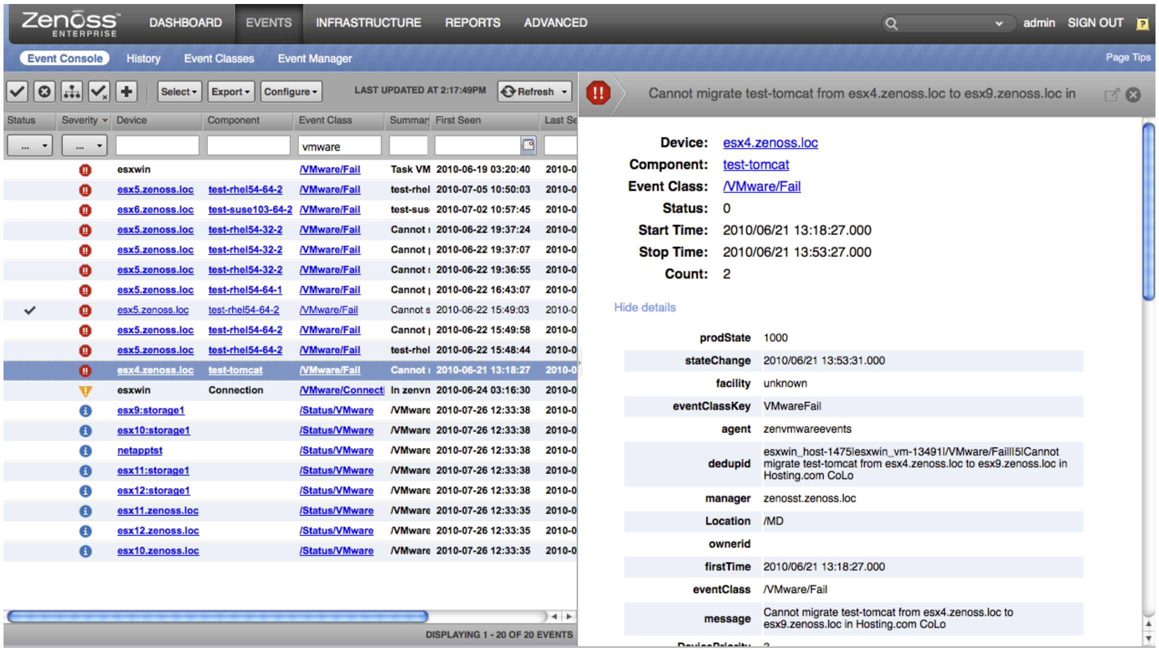
Figure 63.1. VMware Events (Event Console)

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.

Zenoss 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.



LAST UPDATED AT 2:17:49PM

Status	Severity	Device	Component	Event Class	Summar	First Seen	Last Se
...	...			vmware			
	!!	esxwin		/VMware/Fail	Task VM	2010-06-19 03:20:40	2010-0
	!!	esx5.zenoss.loc	test-rhel54-64-2	/VMware/Fail	test-rhel	2010-07-05 10:50:03	2010-0
	!!	esx6.zenoss.loc	test-suse103-64-2	/VMware/Fail	test-sus	2010-07-02 10:57:45	2010-0
	!!	esx5.zenoss.loc	test-rhel54-32-2	/VMware/Fail	Cannot	2010-06-22 19:37:24	2010-0
	!!	esx5.zenoss.loc	test-rhel54-32-2	/VMware/Fail	Cannot	2010-06-22 19:37:07	2010-0
	!!	esx5.zenoss.loc	test-rhel54-32-2	/VMware/Fail	Cannot	2010-06-22 19:36:55	2010-0
	!!	esx5.zenoss.loc	test-rhel54-64-1	/VMware/Fail	Cannot	2010-06-22 16:43:07	2010-0
✓	!!	esx5.zenoss.loc	test-rhel54-64-2	/VMware/Fail	Cannot s	2010-06-22 15:49:03	2010-0
	!!	esx5.zenoss.loc	test-rhel54-64-2	/VMware/Fail	Cannot	2010-06-22 15:49:58	2010-0
	!!	esx5.zenoss.loc	test-rhel54-64-2	/VMware/Fail	test-rhel	2010-06-22 15:48:44	2010-0
	!!	esx4.zenoss.loc	test-tomcat	/VMware/Fail	Cannot	2010-06-21 13:18:27	2010-0
	!	esxwin	Connection	/VMware/Connecti	In zenvn	2010-06-24 03:16:30	2010-0
	i	esx9:storage1		/Status/VMware	/VMware	2010-07-26 12:33:38	2010-0
	i	esx10:storage1		/Status/VMware	/VMware	2010-07-26 12:33:38	2010-0
	i	netapptst		/Status/VMware	/VMware	2010-07-26 12:33:38	2010-0
	i	esx11:storage1		/Status/VMware	/VMware	2010-07-26 12:33:38	2010-0
	i	esx12:storage1		/Status/VMware	/VMware	2010-07-26 12:33:38	2010-0
	i	esx11.zenoss.loc		/Status/VMware	/VMware	2010-07-26 12:33:35	2010-0
	i	esx12.zenoss.loc		/Status/VMware	/VMware	2010-07-26 12:33:35	2010-0
	i	esx10.zenoss.loc		/Status/VMware	/VMware	2010-07-26 12:33:35	2010-0

DISPLAYING 1 - 20 OF 20 EVENTS

Cannot migrate test-tomcat from esx4.zenoss.loc to esx9.zenoss.loc in

Device: esx4.zenoss.loc
Component: test-tomcat
Event Class: /VMware/Fail
Status: 0
Start Time: 2010/06/21 13:18:27.000
Stop Time: 2010/06/21 13:53:27.000
Count: 2

prodState	1000
stateChange	2010/06/21 13:53:31.000
facility	unknown
eventClassKey	VMwareFail
agent	zenvmwareevents
dedupid	esxwin_host-1475lesxwin_vm-13491l/VMware/Faillll5lCannot migrate test-tomcat from esx4.zenoss.loc to esx9.zenoss.loc in Hosting.com CoLo
manager	zenosst.zenoss.loc
Location	/MD
ownerid	
firstTime	2010/06/21 13:18:27.000
eventClass	/VMware/Fail
message	Cannot migrate test-tomcat from esx4.zenoss.loc to esx9.zenoss.loc in Hosting.com CoLo

Figure 63.2. Event Details

63.1.1.1. 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 Zenoss. 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.

63.2. Prerequisites

To implement this ZenPack, you must install:

- OpenSSL development package, Version 0.9.7 or higher
- VMware vSphere CLI (as described in the section titled Installing Prerequisite Libraries).

Prerequisite	Restriction
Zenoss Version	Zenoss Version 3.0 or higher
Required ZenPacks	ZenPacks.zenoss.ZenVMware, ZenPacks.zenoss.StorageBase, ZenPacks.zenoss.DynamicView
VMware VI API	Compatible with ESX Server 3.5, VirtualCenter Server 2.5, and ESX Server 3i. It is not explicitly compatible with ESX Server 3.0.x or VirtualCenter 2.0.x, or any previous versions.

Table 63.1. VMware Prerequisites

Warning

If the time on the monitored VC/ESX server is too far from the time on the box where the **zenvmwareperf** daemon is running, the daemon will not collect any data.

63.2.1. Installing Prerequisite Libraries

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

VMware hosts (as seen in the /Devices/VMware/*EndpointName*/Hosts area) must be resolvable in DNS to obtain performance information from the `resxtop` command.

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 Zenoss 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:

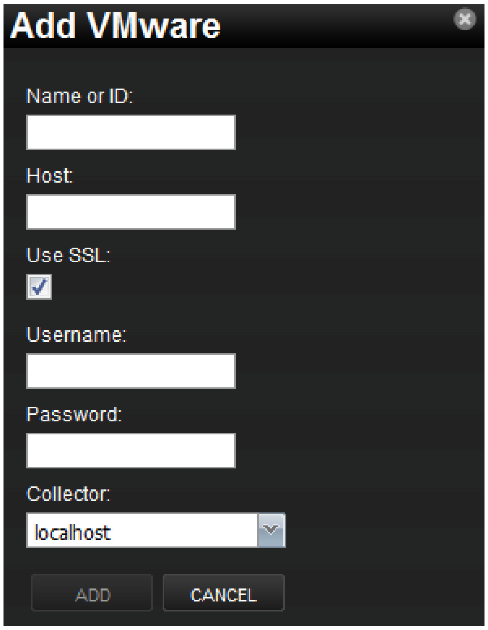
   ```
   $ZENHOME/ZenPacks/Ze*ZenVMware*/Z*/z*/Z*/libexec/check_esxtop --server=myEsxserver \
   --user=userOnRemoteEsxServerAllowedToUseEsxTop --password=password --showvms
   ```

63.3. Enable Monitoring

Follow these steps to begin monitoring your VMware servers.

1. From Infrastructure > Devices, select Add VMware Infrastructure from .

   The Add VMware dialog appears.

   Use SSL: ☑

   Collector: localhost

   Figure 63.3. 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.

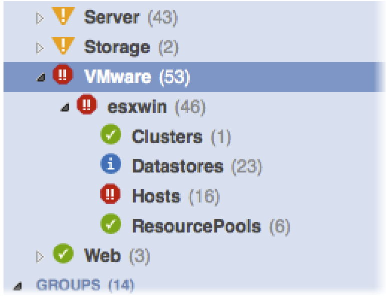
   Zenoss begins modeling the VMware infrastructure. It places the information in the device hierarchy under `/Devices/VMware/ID`, where ID is the value of the ID field you entered during setup.

63.4. Viewing VMware Devices

Zenoss 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.



- Server (43)
- Storage (2)
- VMware (53)
  - esxwin (46)
    - Clusters (1)
    - Datastores (23)
    - Hosts (16)
    - ResourcePools (6)
- Web (3)

GROUPS (14)

Figure 63.4. VMware Device Classes

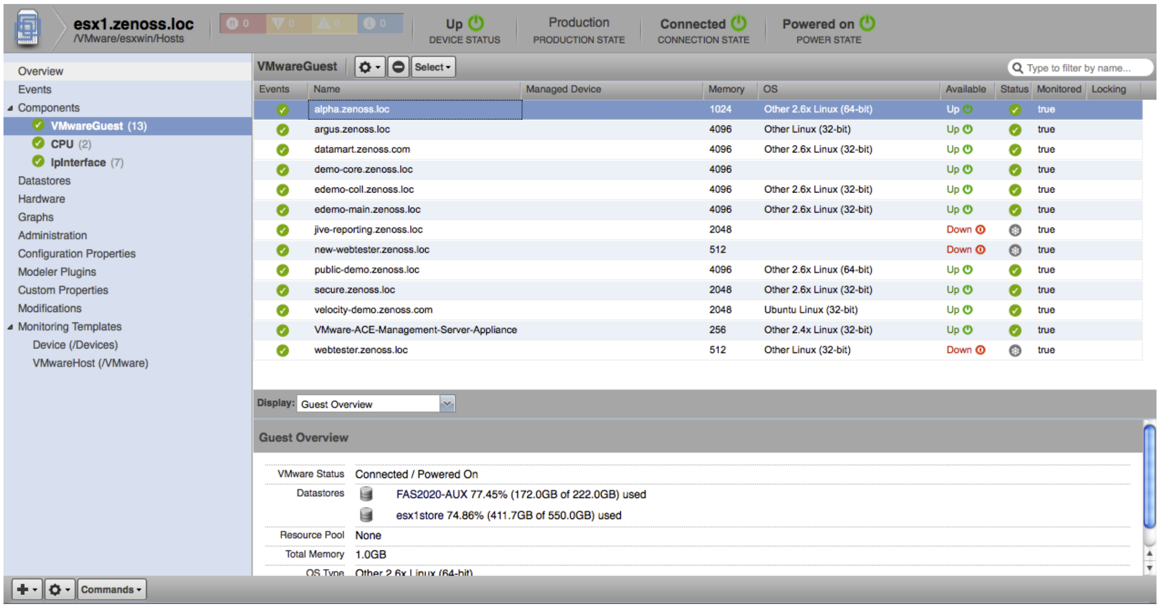
If the SDK endpoint is an individual ESX server, then the Clusters organizer will be empty. (A VMware cluster is a concept external to an individual host.)

63.5. 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.



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		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	

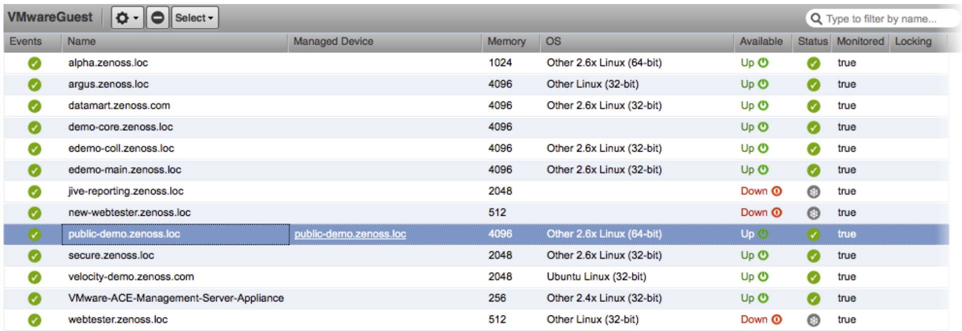
Guest Overview

VMware Status	Connected / Powered On
Datastores	FAS2020-AUX 77.45% (172.0GB of 222.0GB) used esx1store 74.86% (411.7GB of 550.0GB) used
Resource Pool	None
Total Memory	1.0GB
OS Type	Other 2.6x Linux (64-bit)

Figure 63.5. 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 Zenoss, 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.



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	

Figure 63.6. Virtual Guest Devices - Managed Device

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.

63.6. Enabling Data Collection Using resxtop

Follow these steps to enable gathering of VMware host and guest statistics.

63.6.1. Gathering VMware Host Statistics

By default, data collection using `resxtop` statistics is disabled. To enable it:

1. From the Zenoss 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

63.6.2. Gathering VMware Guest Statistics

By default, data collection using `resxtop` statistics is disabled. To enable it:

1. From the Zenoss 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

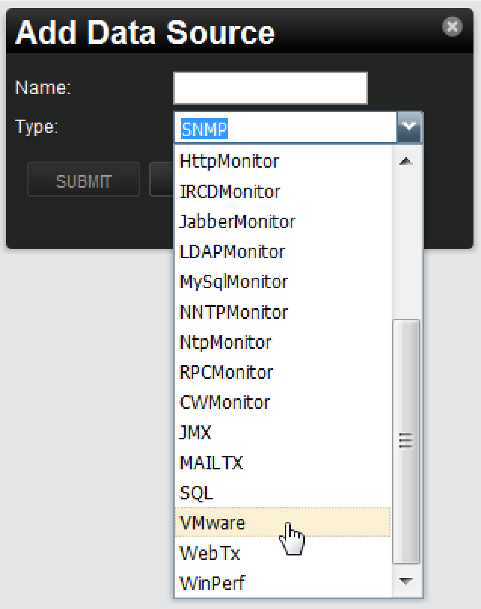
63.7. Adding a Custom Metric

In Zenoss, 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 + to add a data source.

   The Add Data Source dialog appears.



Add Data Source

Name:

Type: SNMP

Figure 63.7. 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 Zenoss, 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.

63.8. 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

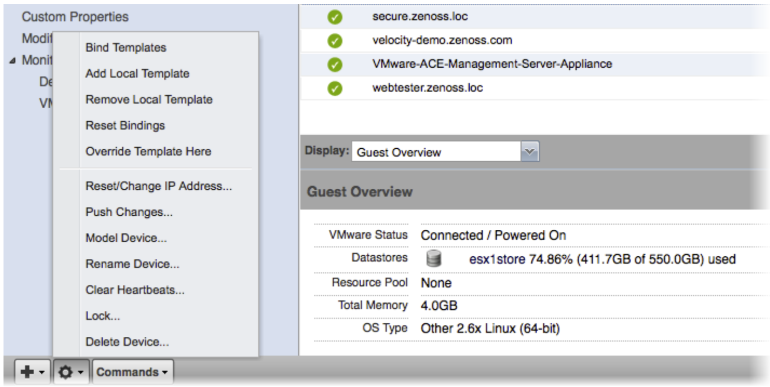
  Note

  Alternatively, as user zenoss, enter the following commands to stop and then restart these Zenoss 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.



secure.zenoss.loc
velocity-demo.zenoss.com
VMware-ACE-Management-Server-Appliance
webtester.zenoss.loc

Display: Guest Overview

Guest Overview

VMware Status	Connected / Powered On
Datastores	esx1store 74.86% (411.7GB of 550.0GB) used
Resource Pool	None
Total Memory	4.0GB
OS Type	Other 2.6x Linux (64-bit)

Figure 63.8. Push Changes

63.9. Daemons

Type	**Name**
Modeler	**zenvmwaremodeler**
Performance Collector	**zenvmwareperf**

Type	Name
Event Collector	**zenvmwareevents**

Table 63.2. Daemons

63.9.1. 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`

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.

Table 63.3. Daemons

- `zenvmwareevents`

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.

Table 63.4. Daemons

Chapter 64. WebSphere Application Server

64.1. About

The WebSphere monitoring feature allows Zenoss to monitor IBM WebSphere Application Servers (WAS).

64.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.ZenWebTx 2.5 or higher, ZenPacks.zenoss.WebsphereMonitor

Table 64.1. WebSphere Prerequisites

64.3. Enable Monitoring

64.3.1. 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.

64.3.2. Configure Zenoss

1. Navigate to the device or device class under the `/Devices/Server/Tomcat` device class in the Zenoss interface.
   - If applying changes to a device class:
     a. Select the class in the devices hierarchy.
     b. Click **Details**.
     c. Select Configuration Properties.
   - If applying changes to a device:
     a. Click the device in the device list.
     b. Select Configuration Properties.
2. Edit the appropriate configuration properties for the device or devices.

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.

Property	Description
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.

Table 64.2. WebSphere Configuration Properties

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.

64.4. 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.

64.5. Daemons

Type	**Name**
Performance Collector	**zenwebtx**

Table 64.3. Daemons

Chapter 65. Web-Based Synthetic Transactions

65.1. About

The ZenWebTx ZenPack allows you to test the availability and performance of Web sites by performing some of the same activities performed by your user community. You create one or more tests that mimic user actions in a Web browser. Zenoss then performs these tests periodically, creating events when a test fails or exceeds a time threshold.

Additionally, Zenoss 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

ZenWebTx 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 ZenWebTx then translates into Twill commands. The **zenwebtx** daemon processes the Twill commands periodically, recording data and creating events as appropriate.

65.1.1. Data Points

Data produced by any Zenoss 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 Zenoss. For more information about the extract and printTimer twill commands, refer to the appendix titled Appendix A, *twill Commands Reference*.

ZenWebTx supports using XPath queries to extract data from XML documents. For more information about this feature, refer to the appendix in this guide titled Appendix A, *twill Commands Reference*.

65.1.2. Event Generation

There are several situations for which ZenWebTx will create events in Zenoss. These events use the component and event class specified on the Data Source tab. These situations are:

- ZenWebTx 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.

65.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.ZenWebTx

Table 65.1. Web Transactions Prerequisites

65.3. Enable Monitoring

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:

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, Zenoss 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.

Table 65.2. WebTx Data Source Options

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.

Enter information or make selections:

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}`. For more information on TALES expressions, refer to the Appendix in the Administratrion Guide titled TALES Expressions.
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 Zenoss 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**, Zenoss 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 Zenoss 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. Zenoss does not save the replaced information. See the Section 65.4, “Creating twill Commands” section for more information about twill commands.

Table 65.3. WebTx Script Settings

Note

If you provide values for Initial User, Initial Password, and Initial Authentication Realm, Zenoss 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.

7. Click **Save** to save the data source.

65.4. Creating twill Commands

ZenWebTx uses a language called 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. The full range of available commands is described in the appendix Appendix A, *twill Commands Reference*.

65.4.1. Creating twill Commands from TestGen4Web

The TestGen4Web Firefox add-on allows you to record browser sessions. ZenWebTx can take these sessions and convert them to twill, creating a starting point for developing ZenWebTx data sources.

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 ZenWebTx data source in Zenoss, 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.

65.4.2. 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 Appendix A, *twill Commands Reference* describes in detail the commands that you can use. 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.

Invoke twill-sh with:

```
> PYTHONPATH=$ZENHOME/Products/ZenWebTx/lib
$ZENHOME/Products/ZenWebTx/bin/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:

```
> PYTHONPATH=$ZENHOME/Products/ZenWebTx/lib
$ZENHOME/Products/ZenWebTx/bin/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.

65.5. Monitoring through Proxy Servers

ZenWebTx can access Web servers through HTTP proxy servers and non-authenticating HTTPS proxy servers.

To configure ZenWebTx 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.

65.5.1. 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/
```

65.5.2. Testing the Proxy Setup

You can test the proxy setup by using the twill-sh tool. twill-sh is an interpreter shell for the twill scripting language, which is used to define `WebTx` data sources.

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:

   ```
   PYTHONPATH=$PYTHONPATH:\
   $ZENHOME/ZenPacks/ZenPacks.zenoss.ZenWebTx/ZenPacks/zenoss/ZenWebTx/lib:\
   $ZENHOME/ZenPacks/ZenPacks.zenoss.ZenWebTx/ZenPacks/zenoss/ZenWebTx/bin/twill-sh
   ```

3. Try to retrieve a URL through HTTP or HTTPS. For example, to retrieve the Zenoss 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
   ```

65.6. Daemons

Type	**Name**
Performance Collector	**zenwebtx**

Table 65.4. Daemons

Chapter 66. Windows Performance

66.1. About

ZenWinPerf is a ZenPack that allows performance monitoring of Windows servers without an intermediary Windows server doing the data collection. ZenWinPerf provides the `WinPerf` Data Source, which 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.

Name	Description
zenwin	Watches Windows services and reports on status.
zeneventlog	Watches Windows Event Log and generates events.
zenwinperf	Collects Perfmon performance data.
zenmodeler	This models Windows devices and has both SNMP and WMI support.

Table 66.1. Windows Monitoring Daemons

66.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.3 or higher
Required ZenPacks	ZenPacks.zenoss.WinModelerPlugins, ZenPacks.zenoss.ZenWinPerf
Supported OS Versions	Windows XP, Windows 2000, Windows 2003, Windows Vista, Windows 2008

Table 66.2. Windows Performance Monitoring Prerequisites

66.3. Enable Monitoring

66.3.1. Defining Windows Credentials

A connection to a Windows device cannot be established without a valid set of credentials. The configuration properties `zWinUser` and `zWinPassword` can be set per device or for an entire device class.

Tip

The user needs to be a member of the local administrators or of the domain administrators group unless the steps in Section 66.8, “Configuring a Standalone Windows Device for a Non-Administrative Account” are followed.

To set these configuration properties:

1. Navigate to the device or device class in the Zenoss interface.
    - If applying changes to a device class:
        a. Select the class in the devices hierarchy.
        b. Click **Details**.
        c. Select Configuration Properties.
    - If applying changes to a device:

   a. Click the device in the device list.
   b. Select Configuration Properties.

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

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.

Table 66.3. Windows Performance Configuration Properties

3. Click Save to save your changes.

66.3.2. Add Devices in Zenoss

The ZenWinPerf ZenPack includes a `/Device/Server/Windows/WMI` class that has several new 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.

66.4. Monitor Other Performance Counters

To create your own `WinPerf` data sources, follow these steps:

1. Navigate to either a new or an 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 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. If you wish you can test the counter by entering a device id in the Test Device field and clicking the Test button.

66.5. Testing Connections from Windows

This procedure verifies that the username/password combination are correct, and that there is no firewall blocking the connection.

1. Run the **wbemtest** command.
2. Click on the Connect… button.
3. In the Namespace field, enter:

```
\\HOST\root\cimv2
```

4. Enter login information in the User and Password fields.
5. Click on the Query field.
6. Enter the following to return a dialog with a list of services on the device.

```
select * from win32_service
```

66.6. Testing Connections from Zenoss

This procedure verifies that the username/password combination are correct, and that there is no firewall blocking the connection. Since this is done from the Zenoss server, this test is a better approximation of how successful Zenoss will be in connecting to the Windows device.

As the zenoss user on the Zenoss server:

```
wmic -U 'user' //device 'select * from Win32_computerSystem'
```

The **wmic** command will then prompt you for the password.

Note

This procedure is only valid for Zenoss 2.3 or greater.

66.7. Modify Registry Settings for Firewalls in Secure Environments

Note

This procedure is only applicable for environments with firewalls and so most users will not need this step.

DCOM dynamically allocates one port per 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.

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`

Table 66.4. Firewall and Registry Settings for DCOM

These registry settings must be established in addition to all firewall settings.

66.8. Configuring a Standalone Windows Device for a Non-Administrative Account

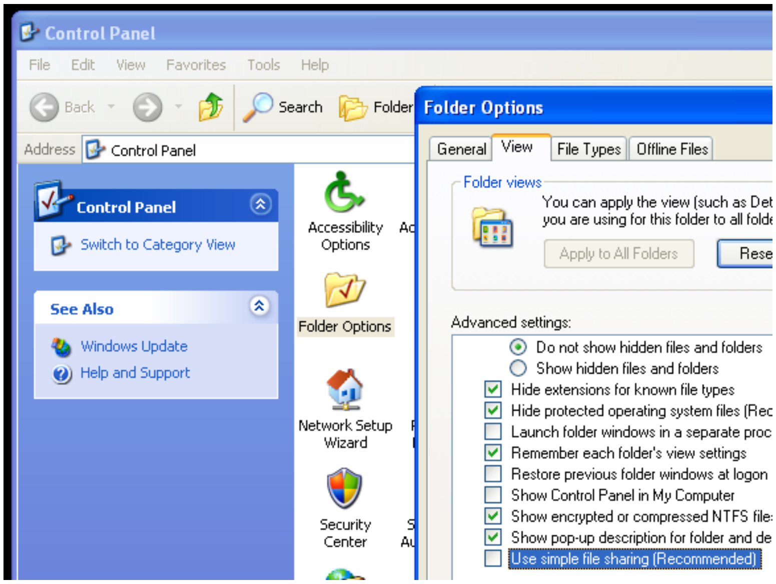
Monitoring Windows devices normally requires an account with administrator-level privileges. For the Zenoss user who wants to use a non-administrative account, several additional configuration steps must be performed on each Windows device, or by using a Group Policy.

Zenoss uses the Windows Management Instrumentation (WMI) feature to collect Event Log and Service information in the Core edition and modeling information when using the Enterprise edition. In the Enterprise edition, 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, navigate to the bottom until you see the Use simple file sharing (Recommended) option, and then disable it.



You can apply the view (such as Det
you are using for this folder to all folde

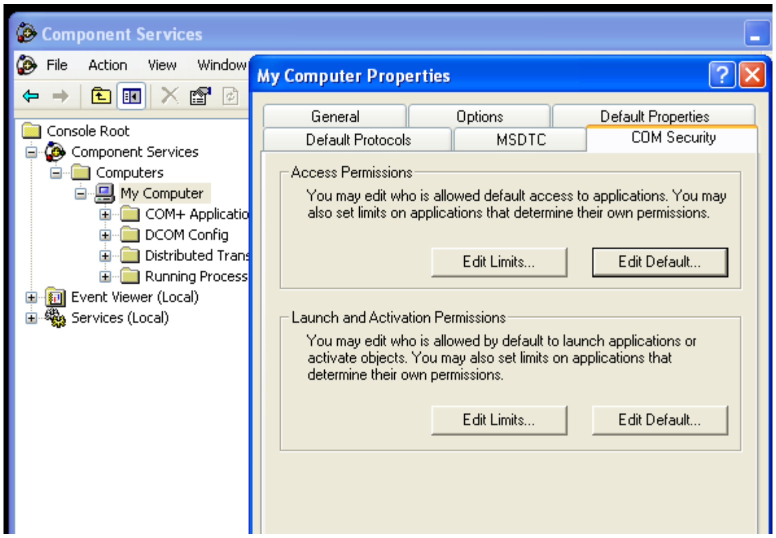
Advanced settings:

- ● Do not show hidden files and folders
- ○ Show hidden files and folders
- ☑ Hide extensions for known file types
- ☑ Hide protected operating system files (Rec
- □ Launch folder windows in a separate proc
- ☑ Remember each folder's view settings
- □ Restore previous folder windows at logon
- □ Show Control Panel in My Computer
- ☑ Show encrypted or compressed NTFS file
- ☑ Show pop-up description for folder and de
- □ Use simple file sharing (Recommended)

Figure 66.1. 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.



You may edit who is allowed default access to applications. You may also set limits on applications that determine their own permissions.

You may edit who is allowed by default to launch applications or activate objects. You may also set limits on applications that determine their own permissions.

Figure 66.2. 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 .

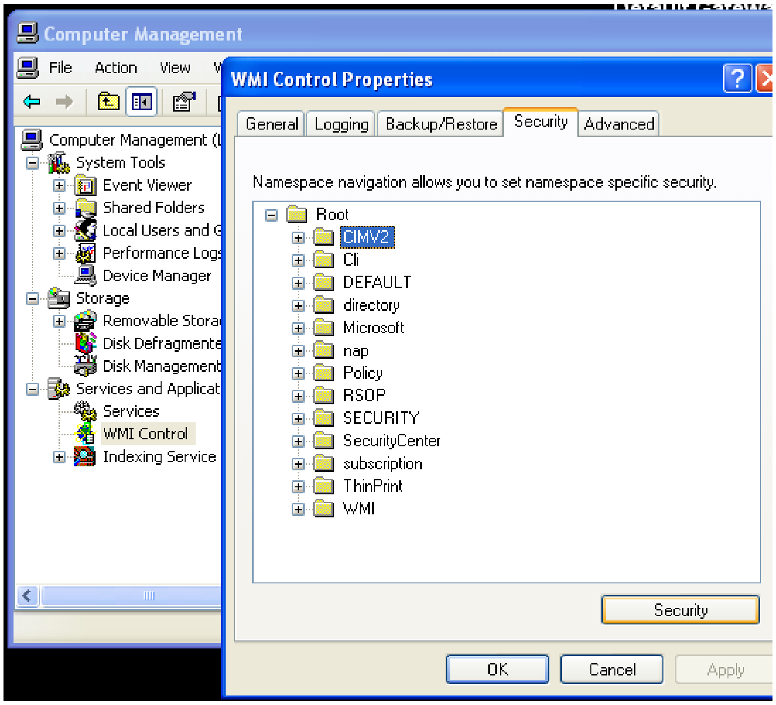
b. In the My Computer Properties dialog box, click the COM Security tab.

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.



Namespace navigation allows you to set namespace specific security.

Figure 66.3. 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 Zenoss 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**.

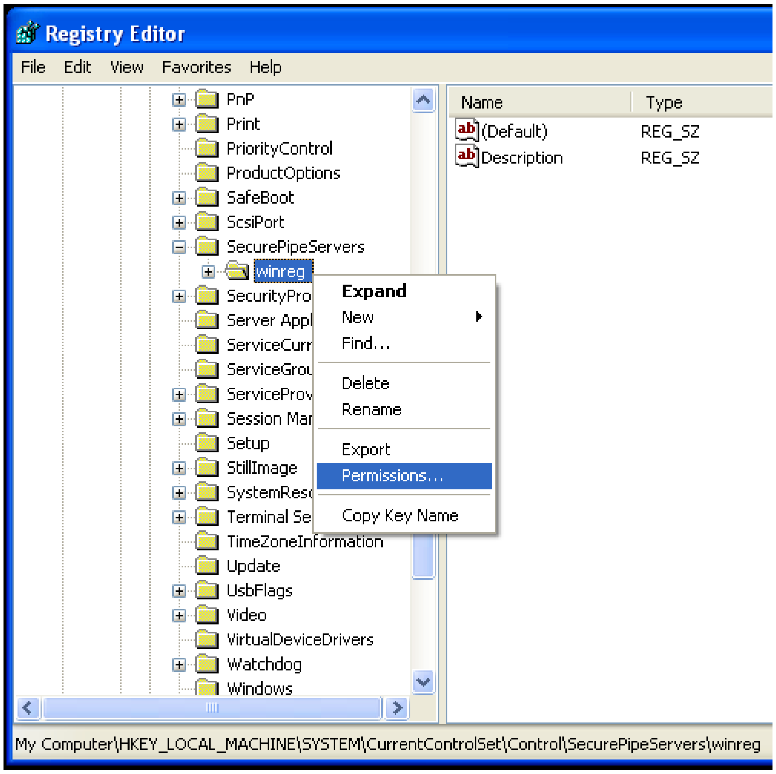


Figure 66.4. regedit and the winreg Key

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 Zenoss 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)
```

Warning

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 Zenoss system:

```
wmic -U '.\zenossmon' //myhostname 'SELECT Name FROM Win32_Service'
```

66.9. Tuning Collector Daemon Performance

ZenWinPerf 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.

Property	Setting
zWinPerfCycleSeconds	This is how frequently (in seconds) **zewinperf** data sources are collected. By default this is set to 300 seconds.

Table 66.5. zenwinperf Daemon Configuration Properties

66.10. 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
```

Chapter 67. Zenoss Global Dashboard

67.1. About

The Zenoss Global Dashboard is a standalone Web server that collects event and heartbeat data from the monitored Zenoss servers and aggregates them into a single view. Several portlets from the standard Zenoss dashboard are available:

- **Device Issues** - A list of all devices with serious events. The Server column displays the Zenoss server that monitors that device.
- **Zenoss Sub-Systems** - A list of monitored Zenoss instances. An event rainbow is displayed for each instance, showing a summary of active events.
- **Zenoss Issues** - A list of heartbeat issues from monitored Zenoss instances. Refer to the Zenoss Administration Guide for instructions on how to handle these events.

Note

ZenGlobe is a standalone Web server. It is not a ZenPack.

67.2. Prerequisites

Prerequisite	Restriction
Other requirements	The Python **setuptools** package is required.

Table 67.1. Zen Global Dashboard Prerequisites

67.3. Configuration

67.3.1. Install the ZenGlobe Web Server

Follow these steps to download and install the Zenoss Global Dashboard:

1. Download the latest version of the Zenoss Global Dashboard.
2. Extract the tarball and change to the created directory using the following commands:

   ```
   tar xzf ZenGlobe-2.1.tar.gz
   cd ZenGlobe-2.1
   ```

3. Install ZenGlobe.

   ```
   sudo python setup.py install
   ```

4. Prior to starting up the first time, ZenGlobe needs to know the port it should bind to and the Zenoss instance it should use for authentication. Run:

   ```
   sudo zenglobe configure
   ```

   Enter the port to which you want ZenGlobe to bind. Make sure you have nothing else listening at that port.

   When asked, enter the hostname of a running Zenoss instance that you want ZenGlobe to use for authentication. You can change this setting later, but in order to log in to ZenGlobe the first time, you will need to use the username and password of a user from this Zenoss instance. Anyone with a login to this instance will be able to view the ZenGlobe dashboard, but only the `admin` user will be able to edit settings.

5. Start ZenGlobe using the command:

   ```
   sudo zenglobe start
   ```

6. Check to make sure that ZenGlobe has started by accessing from your browser:

   ```
   http://[ZenGlobe machine hostname]:[port]/
   ```

   You should now see a ZenGlobe login screen.

67.3.2. Configure Remote Zenoss for Monitoring

For security reasons, ZenGlobe must be configured to log in to the remote Zenoss instances from which it gathers data. By default, this is set to be `zenglobe:zenglobe`; however, it is a good idea to reconfigure ZenGlobe to use a more secure username and password combination.

1. In a browser, navigate to the Zenoss instance you wish to monitor, click Settings in the left navigation pane.
2. Select the Users tab.
3. From the table menu, select Add New User. Enter the username that you want ZenGlobe to use to log in to all Zenoss instances (e.g. `zenglobe`). You may leave the Email field blank.
4. Click the OK button to save your changes.

67.3.3. Configure ZenGlobe to Monitor Remote Zenoss Instances

1. Log in to the Zenoss Global Dashboard as the `admin` user.

   Note

   Only the `admin` user can modify ZenGlobe options.

2. Click the Configure... link in the top bar. The configuration box will slide down.

   The options in this configuration box are as follows:

Name	Description
Zenoss Servers	The list of hostnames of the Zenoss instances ZenGlobe will monitor.
Remote Login	The user name and password ZenGlobe will use to access the remote Zenoss instances. By default, it is set to `zenglobe:zenglobe`. Follow the instructions in Section 67.3.2, “Configure Remote Zenoss for Monitoring” to set up matching users on each Zenoss instance to be monitored.
URL Template	The template ZenGlobe will use to build the URL by which it accesses monitored Zenoss instances. If you run your Zenoss instances on a different port, or serve them behind Apache with rewritten URLs, you will need to update this value to reflect that change.
Authentication Server	The Zenoss instance against which ZenGlobe authenticates. You may also reset the port and authentication server using the same command line option you used when initially configuring ZenGlobe.

Table 67.2. Zen Global Dashboard Configuration Options

67.4. Viewing a Remote Zenoss Instance

The drop-down list on the extreme left of the top bar can be used to view monitored Zenoss instances from within ZenGlobe. Select the hostname of an instance from the list and then log in to the remote instance. You may return to the ZenGlobe dashboard at any time by selecting it from the same drop-down list.

67.5. Ending a Session

Click Logout in the top bar to end your ZenGlobe session.

Chapter 68. ZenOperator Role

68.1. About

The ZenOperatorRole ZenPack creates a new role (`ZenOperator`) suitable for use in Zenoss. For more information about using this role, please see the Zenoss Administration Guide section titled "Roles" in the chapter titled "Managing Users."

68.2. Prerequisites

Prerequisite	Restriction
Zenoss Version	Zenoss Version 2.2 or higher
Required ZenPacks	ZenPacks.zenoss.ZenOperatorRole

Table 68.1. Zen Operator Role Prerequisites

Appendix A. twill Commands Reference

A.1. About

twill is the language used by ZenWebTx to simulate user actions in a Web browser and to test pages retrieved by the simulation. The following sections list the twill commands available for use in ZenWebTx data sources.

Note

For detailed information about ZenWebTx, see the chapter titled Chapter 65, *Web-Based Synthetic Transactions*.

Some twill commands produce text output (see the section titled Section A.4, “Display”). These commands do not affect the execution of tests by ZenWebTx, and are useful in testing and debugging ZenWebTx data sources.

To see the output of commands that produce text output, click **Test Twill Commands** on the Script page of a ZenWebTx data source.

Twill commands are divided among the following categories:

- Browsing
- Assertions
- Display
- Forms
- Cookies
- Debugging
- Other commands

A.2. 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.

A.3. 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.

A.4. 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.

A.5. 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.

A.6. 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.

A.7. 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).

A.8. 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.

A.9. 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
```

A.10. ZenWebTx Extensions to twill

ZenWebTx adds several commands to the standard twill vocabulary.

A.10.1. 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 Zenoss, 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 Zenoss thresholds and graph definitions.

A.10.2. 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) 2005-2009 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 Zenoss 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.

A.10.3. twillxpathextract

Zenoss 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 used to retrieve the data.

When applied to an XML document, the XPath expression must return a numeric value. This value is then assigned to the dataName data point.

A.10.4. 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.