Zenoss®

Transforming IT Operations

Zenoss Service Dynamics Analytics Installation and Administration

Release 4.3.1

Zenoss, Inc.

www.zenoss.com

Zenoss Service Dynamics Analytics Installation and Administration

Part Number: 22-032014-4.3-v15

Chapter 1. About Analytics

1.1. Introduction

Zenoss Service Dynamics Analytics and Optimization ("Analytics") provides enterprise reporting for Zenoss Service Dynamics Resource Management ("Resource Manager") instances, enabling improved business intelligence and knowledge management. With Analytics, you have access to:

- Catalog reports that provide insight into your entire infrastructure
- Extensible dashboards that provide operational and executive views
- A robust, ad-hoc reporting feature to facilitate self-management
- A built-in report scheduler
- A wide range of report export formats, including PDF, CSV, Excel, Word, HTML, and JPG.

Analytics reports and features are accessed through an integrated interface. Reports are distributed via email.

Since Jaspersoft provides the back-end technology used to provide the reporting capability, consult the *JasperReports Server User Guide, Release 5.2* for detailed information about the user interface of Zenoss Analytics.

1.2. Analytics Architecture

Analytics combines extract, transform, and load (ETL) procedures with a data warehouse and reporting tools to process report data.

Resource Manager uses three data sources that are key for reporting:

- Zope object database (ZODB)
- MySQL event database (EventDB)
- Round robin databases (RRD), comprising a distributed collection of performance metrics

One component of Analytics, the ZenETL ZenPack, extracts data from the Resource Manager data sources to build a data warehouse used for enterprise reporting. The data warehouse schema is driven by meta data provided by ZenETL, providing a flexible platform for report creation.

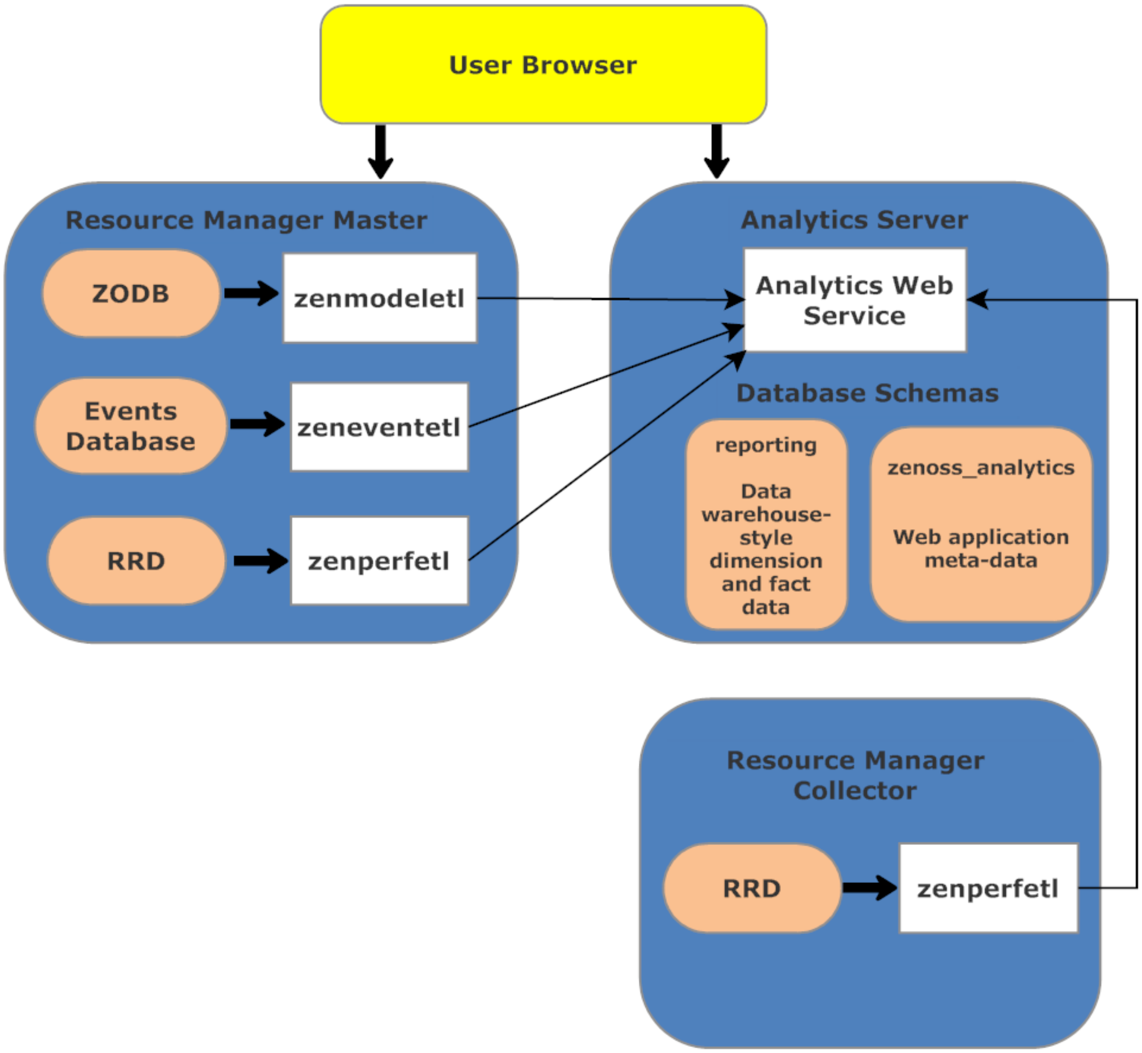
ETL is not a continuous process; it extracts and processes data periodically (according to user-defined parameters) in *batches*. Model, event, and performance data are independently extracted.

On the Resource Manager server, three daemons run to export data from the Resource Manager server to the Analytics server:

- **zeneventetl** - Extracts event information, which is then sent to the Analytics server for subsequent transform and load.
- **zenmodeletl** - Extracts ZODB model information, which is then sent to the Analytics server for subsequent transform and load.
- **zenperfetl** - Extracts RRD performance information, which is then sent to the Analytics server for subsequent transform and load. This daemon also runs on each collector. Typically, this is run on the Resource Manager server and at each remote collector.

All RRD Data Points are available for performance data extraction. You can select which data points are extracted by assigning them an alias.

Figure 1.1. Analytics Components



Chapter 2. Installing and Configuring Analytics

The following sections provide information and procedures to help you install and set up Analytics:

- Installation Considerations
- Prerequisite Tasks and Conditions
- Installing the Oracle Java JDK
- Installing the Zenoss DataStore on the Analytics Server
- Installing the Analytics Server
- Installing the ZenETL ZenPack on Resource Manager
- Configuring Analytics
- Removing an Analytics Installation

2.1. Installation Considerations

The Analytics system creates two databases:

- **zenoss_analytics** - Stores report design, execution information, and results of the reports, including PDFs. You can also access previous reports here.
- **reporting** - Warehouses all data retrieved from Resource Manager. Also referred to as the "etl database."

  The reporting database can become large, and load can be significant. For large-scale installations, Zenoss recommends you install the zenoss_analytics database and the Analytics application on one machine, and the reporting database on a separate machine. All of the Analytics servers should be separate from the Resource Manager server.

2.2. Required Software and Conditions

- To run Analytics, you must install:
  - Oracle Java SE 6 Java Development Kit (JDK), this is different than the Java Runtime Environment (JRE) that was installed as part of the Resource Manager installation.

    ### Note

    Do not use Java SE 7 or the OpenJDK. They are not compatible with this version of Analytics.

  - Zenoss DataStore on the Analytics Server
  - The following ZenPacks (included with Resource Manager 4.2.x and Zenoss Enterprise 4.1.1):
    - Distributed Collector
    - vCloud
    - VMware

- Analytics requires that UID 1337 be available. You can verify its availability with the command:

  ```
  getent passwd 1337
  ```

- The Analytics server and any associated Resource Manager instances must be running Network Time Protocol (NTP) so that their clocks remain synchronized. (For more information about NTP, visit http://www.ntp.org/.)

2.2.1. Compatibility and Support

- Analytics 4.3 is compatible with Service Dynamics 4.1.1 + RPS and later versions.
- You must use matched versions (first 3 digits of version number) of Analytics and the ZenETL ZenPack. For example, you cannot use ZenETL 4.2.4 with Analytics 4.3.1, or ZenETL 4.3.1 with Service Dynamics 4.2.4.
- You must be using a 64-bit system running RHEL 5/CentOS 5 or RHEL 6/CentOS 6.
- The `zenperfetl` daemon that runs on remote collectors makes Web services calls to the Analytics server, and thus must be able to access the server's HTTP address.
- The system is not compatible with Security-Enhanced Linux (SELinux) in enforcing mode. You must disable SELinux to run Analytics.

  To disable enforcing mode in SELinux:

  1. Edit the /etc/selinux/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.

2.2.2. Recommendations

- You should install Analytics and its prerequisites on a server separate from the Resource Manager server.
- Analytics is a data warehouse application. Like other database applications, it benefits from fast disc subsystems, fast processors, and abundant RAM.

  Zenoss recommends that you meet these minimum system requirements when running Analytics:

  - Dual core processor
  - 8GB RAM
  - 500GB storage

2.2.3. Open Network Ports

The Analytics system comprises multiple components. Communication needs for each of these components differs depending on how you have installed your system (on a single box or multiple boxes, or with or without distributed collectors).

The following table lists Analytics pieces that typically must communicate, and the default network ports that should be open for communication.

From	To	Default Port Numbers
Analytics	Analytics database server	13306
Analytics	Resource Manager	8080
Resource Manager	Analytics	7070
Analytics Web Users	Analytics	7070
`zenperfetl` (on each remote collector)	Analytics application server	7070

Note

If you want to set up Analytics communication to use SSL, see the Configure SSL Communication section.

2.3. Installing the Oracle Java JDK

Analytics requires the Oracle Java 1.6 JDK for installation. It cannot install when just the JRE is installed. To determine if you already have the JDK installed, run the following command:

```
rpm -q jdk
```

If you do not have the JDK installed, perform the following steps:

1. Download the self-installing RPM of Oracle Java SE Development Kit 6u31 from the Java SE 6 Downloads page. The file to download is `jdk-6u31-linux-x64-rpm.bin`.

   ### Note

   Before the file can be downloaded, you must accept the license agreement.

2. Change mode:

   ```
   chmod +x ./jdk-6u31-linux-x64-rpm.bin
   ```

3. Install Oracle Java JDK:

   ```
   ./jdk-6u31-linux-x64-rpm.bin
   ```

Note

The JDK is only needed to install Analytics. Once Analytics is installed and running, you can remove the JDK if you wish.

2.3.1. Setting the JAVA_HOME Environment Variable

After installing the Oracle Java JDK, you must set `$JAVA_HOME` for the root user:

1. Locate the path to your JDK directory. Typically, this is `/usr/java/jdk`*VersionNumber*.

2. Add the following lines to the `/root/.bash_profile` file:

   ```
   export JAVA_HOME=/usr/java/jdkVersionNumber
   export PATH=$JAVA_HOME/bin:$PATH
   ```

3. Test the new path with the command:

   ```
   #echo $JAVA_HOME
   ```

2.4. Installing the Zenoss DataStore on the Analytics Server

Follow these steps to install the Zenoss DataStore on the Analytics server.

Note

Before installing the Zenoss DataStore on systems running Red Hat Enterprise Linux (RHEL), install the Perl Database Interface Module (DBI) with the following command.

```
yum install perl-DBI
```

Install the Zenoss DataStore:

1. Browse to this URL, and then locate the Zenoss DataStore RPM files:

   https://support.zenoss.com

   ### Note

   Contact your Zenoss representative for site login credentials.

2. Install the Zenoss DataStore. As the root user, enter the following command based on the RPM version you are installing:

   ```
   rpm -ivh zends-5.6.16-1.Version.RHEL.x86_64.rpm
   ```

   where *Version* is the build version number and *RHEL* is the RHEL or CentOS version (e.g., el5 or el6)

   ### Note

   Be sure to use version 5.6 of `zends`. Version 5.5 is not supported for Analytics 4.3.1.

3. Create a new /tmp folder in the /opt/zends directory to avoid having zends fill up the default /tmp directory.

   ```
   cd /opt/zends
   mkdir tmp
   chown zenoss:zenoss tmp
   ```

4. Configure the Zenoss DataStore by adding the following lines to the end of the `/opt/zends/etc/zends.cnf` file:

   ```
   [mysqld]
   table_open_cache=16K
   table_definition_cache=16K
   tmp_table_size=2G
   max_heap_table_size=2G
   join_buffer_size=512K
   open_files_limit=200000
   tmpdir=/opt/zends/tmp
   ```

5. In the same `/opt/zends/etc/zends.cnf` file, edit the following lines to reflect the values listed below. Set the innodb_buffer_pool_size to be between 70 and 80 percent of total memory.

   ```
   innodb_adaptive_hash_index=OFF
   innodb_buffer_pool_size=6G
   innodb_log_file_size=1892M
   innodb_log_buffer_size=128M
   ```

6. Run the following commands:

```
service zends start
chkconfig zends on
```

If your Analytics application and the Analytics database are on the same machine, then you are done with the Zenoss DataStore install. However, if the Analytics application and database are on different machines or if you want to connect to a development tool directly, perform the following steps to enable remote access to the Zenoss DataStore:

7. Change to the zenoss user:

```
su - zenoss
```

8. Enter the following command:

```
zends -u root
```

9. Enable remote access to the Zenoss DataStore. In the prompt that appears, enter this series of commands:

```
grant all on *.* to 'root'@'hostname' with grant option;
grant all on *.* to 'reporting'@'hostname' identified by 'reporting';
flush privileges;
```

Note

Whatever value you use here for the password needs to be added to the `/opt/zends/etc/zends.cnf` file that was edited in step 4.

10.If you have a firewall between the Analytics application server and the Analytics database server, be sure of the following:

a. Open up the ports to allow the connections.

b. Do not kill any long running connections.

2.5. Installing the Analytics Server

To install the Analytics server:

1. As the root user, enter the following command based on the RPM version you are installing:

```
rpm -ivh zenoss_analytics-4.3.1.0.0-1.Version.noarch.rpm
```

2. Restart the Zenoss DataStore. (This clears the resource limit values previously set in the `/etc/security/limits.conf` file.)

```
service zends restart
```

3. Modify the database connection setting in the `/etc/zenoss_analytics/zenoss_analytics.conf` file to point to the desired database server or servers. This step is not necessary if the database is on the same server, in which case you should leave it as `localhost`.

```
etl.jdbc.hostname=<remote database server>
```

Note

Be sure to modify the admin account information to match the account or accounts you created when setting up the database.

4. Change to the zenoss user:

```
su - zenoss
```

5. Install the database schema with the command:

   ```
   /opt/zenoss_analytics/bin/upgrade_db.py
   ```

6. Change to the root user:

   ```
   su - root
   ```

7. Enable chkconfig for the Analytics service:

   ```
   chkconfig zenoss_analytics on
   ```

8. Modify the JVM configuration to increase the PermGen space:

   a. Modify `/opt/zenoss_analytics/bin/setclasspath.sh` with the following value for the JAVA_OPTS:

      ```
      # Set JVM options for increased resource requirements of JasperServer
      export JAVA_OPTS="$JAVA_OPTS -Xms128m -Xmx1024m -XX:PermSize=32m
      -XX:MaxPermSize=512m -Xss2m -XX:+UseConcMarkSweepGC
      -XX:+CMSClassUnloadingEnabled"
      ```

9. Start the Analytics service:

   ```
   service zenoss_analytics start
   ```

2.6. Installing the ZenETL ZenPack on the Resource Manager

After installing the Analytics server, install the ZenETL ZenPack on the Resource Manager master. You can install it using the command line or from the Resource Manager interface. See the corresponding directions below:

To install using the command line:

1. As the zenoss user, enter the following command based on the version you are installing:

   ```
   zenpack --install ZenPacks.zenoss.ZenETL-Version-py2.7.egg
   ```

2. Create aliases to the data points on the monitoring templates that you want to bring into Analytics. See the "Running the chkaliases script" for instructions on using the script found in the ZenETL ZenPack.

3. Restart the system:

   a. Enter the following commands:

      ```
      zenoss stop
      ps -eaf | grep $ZENHOME
      ```

   b. For any process that is reported as running, perform a `kill -9` on those processes.

   c. Start the system:

      ```
      zenoss start
      ```

   ### Note

   Optionally, if you are using `daemons.txt` on the master, you should add entries for `zeneventetl`, `zenmodeletl`, and `zenperfetl` to the file. If you are using `collectordaemons.txt` on the collectors, you should add an entry for `zenperfetl` to the file.

When you restart the Resource Manager master, three additional daemons (`zeneventetl`, `zenmodeletl`, and `zenperfetl`) are started. The `zenperfetl` daemon is started on each remote collector.

Be sure to install the latest RPS for your Resource Manager instance so that you get all the latest fixes. The Distributed Collector ZenPack should be at version 2.5.12 or higher otherwise you will be required to use a `daemons.txt` and a `collectordaemons.txt` file to stop `zenmodeletl` and `zeneventetl` on running collectors.

2.7. Configuring Analytics

To configure Analytics, you must:

- Configure the connection to the Analytics server
- Configure zeneventserver settings
- (Optionally) Configure Database credentials
- (Optionally) Configure additional Resource Manager data points to send to Analytics
- (Optionally) Configure SSL communication
- Configure email services

2.7.1. Configure the Connection to the Analytics Server

Follow these steps to configure Resource Manager to connect to the Analytics server:

1. From the Resource Manager interface navigation bar, select Reports > Configuration.

   The Analytics configuration page appears.

2. In the Internal URLs area, enter the URLs that Resource Manager and Analytics will use to communicate directly:

   - **Analytics** - Enter the fully qualified domain name (including port number) for the Analytics server. The default port number is 7070.
   - **Resource Manager** - Enter the fully qualified domain name (including port number) for the Resource Manager server. (By default, this information is pre-populated.)

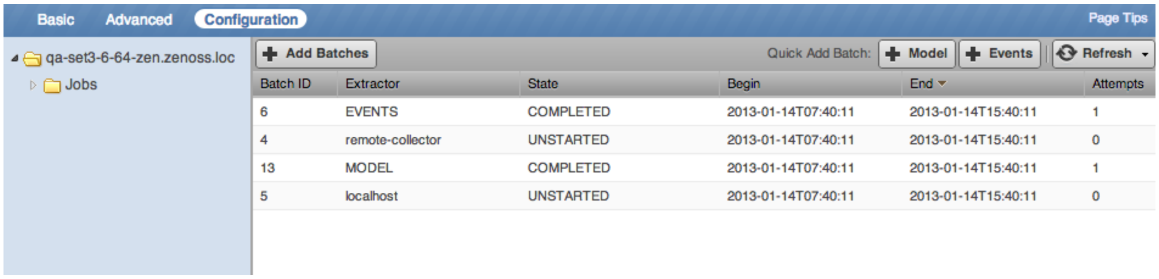
3. In the External URLs area, enter the links that will be displayed to users in each of the applications:

   - **Analytics** - Enter the complete URL, including port number, to Analytics. The default port number is 7070.
   - **Resource Manager** - Enter the complete URL, including port number, to Resource Manager. (By default this information is pre-populated.)

4. Click **Submit**.

   The configuration page redisplays with the currently scheduled report jobs.

Figure 2.1. Add Batches



Batch ID	Extractor	State	Begin	End	Attempts
6	EVENTS	COMPLETED	2013-01-14T07:40:11	2013-01-14T15:40:11	1
4	remote-collector	UNSTARTED	2013-01-14T07:40:11	2013-01-14T15:40:11	0
13	MODEL	COMPLETED	2013-01-14T07:40:11	2013-01-14T15:40:11	1
5	localhost	UNSTARTED	2013-01-14T07:40:11	2013-01-14T15:40:11	0

Note

If this is a new installation, then jobs will not initially appear.

Note

On distributed collector installations, the updated configuration must be pushed to each hub, and then to each collector.

2.7.2. Configure zeneventserver Settings

Resource Manager (Version 4.1 and higher) is configured to store only one day of event occurrences. If event extraction falls behind (meaning event batches are older than 24 hours), then event extraction will not properly extract event times for those batches.

You can increase the number of days of event history that can be extracted, from the Resource Manager interface:

1. Select Advanced, and then select Events from the left panel.

   The Event Configuration page appears.

2. Increase the value of Event Time Purge Interval (in days), up to 7 days.

3. Click **Save**.

   ### Note

   If you increase the Event Time Purge Interval value, then you must also increase the number of file descriptors for the Zenoss DataStore for your Resource Manager instance (see next set of instructions).

To increase the number of file descriptors for the Zenoss DataStore:

1. As the root user, view the current open files:

   ```
   lsof | wc -l
   ```

2. Check the max open files by entering the following command:

   ```
   cat /proc/sys/fs/file-max
   ulimit -Hn
   ulimit -Sn
   ```

3. Restart zends which should reset the files for the current time since zends uses many file descriptors:

   ```
   service zends restart
   ```

4. You can also change the max open files temp by entering the following command:

   ```
   sysctl -w fs.file-max=$(($(cat /proc/sys/fs/file-max) + 25000))
   ```

5. Add the following lines to the `/etc/security/limits.conf` file:

   ```
   root hard nofile 999999
   root soft nofile 999999
   zenoss hard nofile 999999
   zenoss soft nofile 999999
   ```

2.7.3. Configure Database Credentials

When Analytics is installed it configures a JDBC data source for the reporting database. If you have a split application and database install, you need to change the database settings in Analytics. Even if you do not have a split installation, it is a good idea to test the database connection.

To change database settings in Analytics:

1. Log in to the Resource Manager using your Resource Manager user name and password. You must have Manager or ZenManager privileges to perform this task in Analytics. For more information about Roles and User Permissions, see the "Managing Users" chapter in *Resource Manager Administration*.

2. Select **Reports > Advanced**.

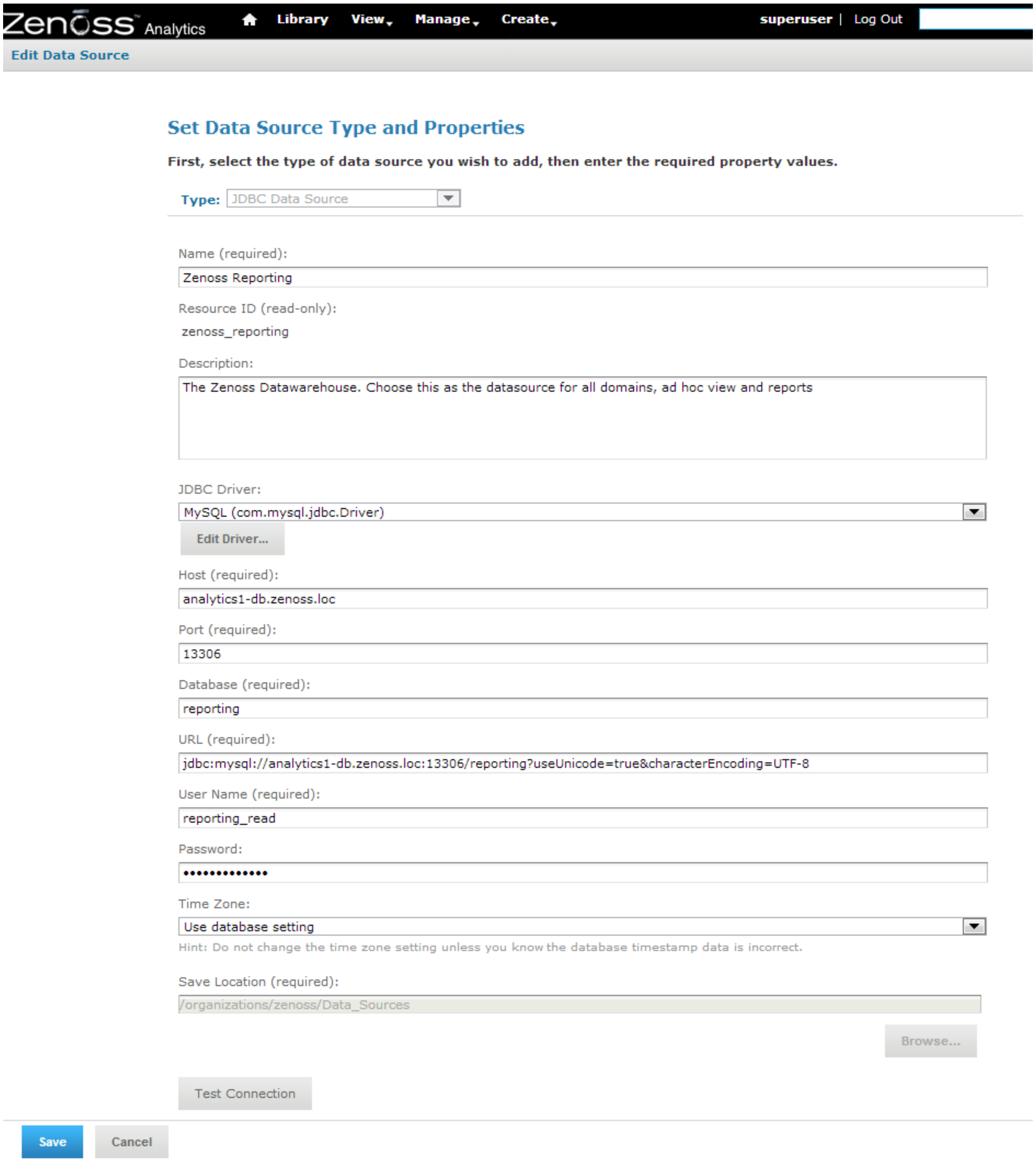
3. On the Analytics Getting Started page, click **View Reports**.

4. On the View drop-down list, select **Repository**.

5. In the Folders area, expand the following folders: **root > Organizations > Zenoss > Data Sources**.

6. Click **Zenoss Reporting** in the Repository area, then select **Edit** from the list of options. The Edit Data Source window is displayed.

Figure 2.2. Edit Data Source window



Set Data Source Type and Properties

First, select the type of data source you wish to add, then enter the required property values.

Type: JDBC Data Source

Name (required): Zenoss Reporting

Resource ID (read-only): zenoss_reporting

Description: The Zenoss Datawarehouse. Choose this as the datasource for all domains, ad hoc view and reports

JDBC Driver: MySQL (com.mysql.jdbc.Driver)

Host (required): analytics1-db.zenoss.loc

Port (required): 13306

Database (required): reporting

URL (required): jdbc:mysql://analytics1-db.zenoss.loc:13306/reporting?useUnicode=true&characterEncoding=UTF-8

User Name (required): reporting_read

Password: •••••••••••••

Time Zone: Use database setting

Hint: Do not change the time zone setting unless you know the database timestamp data is incorrect.

Save Location (required): /organizations/zenoss/Data_Sources

7. After editing, click **Test connection** to verify the changes are valid.

8. Click **Save**.

2.7.4. Export Data Point Data

This section provides information about how Analytics and Resource Manager share performance data, and how to manually modify data points.

Resource Manager uses RRDTool to store performance data. In order for data to be exported for use in the Analytics system, you must create aliases to the data points on the monitoring templates. For more information about templates, see *Zenoss Service Dynamics Resource Management Administration*.

Resource Manager data points represent raw, collected metrics, which are stored in RRD. For example, view the /Server/Linux monitoring template. The ssCpuIdle data point is collected by the ssCpuIdle data source. ssCpuIdle represents the percentage of idle time for all CPUs on this box.

Edit the data point, and you can examine its defined aliases. The cpu__pct alias represents the percentage of CPU used by the box. The alias value is calculated by using the data point and a reverse Polish notation (RPN) expression. Aliases are used to normalize raw data point values across data point types. While it may seem like a good idea to collect lots of data by adding many aliases, you must balance the need for data with the load that it would generate on the system.

You can provide a formula to be used during extraction of data. This allows you to convert data points with different units to the same units during extraction. For example, if you are monitoring disk space on a Unix system (bytes) and a Windows system (bits), you can use a formula to convert both to kilobytes and have them use the same alias.

There are two ways you can add aliases to a data point, which are discussed below:

- Running the chkaliases script
- Manually modifying the data points on the Resource Manager

Running the chkaliases script

To add or remove aliases, run the `chkaliases.zendmd` script contained in the ZenETL ZenPack. This script processes the `aliases.dsv` file, which contains alias definitions. Each useful line of the `aliases.dsv` file should contain an alias definition consisting of the following five values that are pipe-delimited: `device class|template|datapoint|alias|rpn.`

To run the `chkaliases.zendmd` script: can be run on the command-line as the `zenoss` user. The following lists all the possible arguments and options:

1. Log in to the Resource Manager server as the zenoss user:

   ```
   su - zenoss
   ```

2. Navigate to the directory where the script resides by entering the following command (using the actual version number):

   ```
   cd $ZENHOME/ZenPacks/ZenPacks.zenoss.ZenETL-Version-py2.7.egg/ZenPacks/zenoss/ZenETL/etc
   ```

3. Run the script using the following command to see what would be changed:

   ```
   zendmd --script=../bin/chkaliases.zendmd [add | remove]
   ```

   The output of the script produces some codes. Codes in the first column indicate the script is adding (+), modifying the RPN (R), or removing (-) an alias. Codes in the second column indicate an inability to find a device class (c), template (t), datapoint (d), or alias (a). It could also indicate an unwillingness to remove an alias because its defined RPN does not match (r). The (a) and (r) codes only occur during a remove operation.

   The initial `aliases.dsv` file contains alias definitions for the following, expressed as percentages:

   - Device Level CPU usage (cpu__pct)
   - Memory utilization (mem__pct)

- IP interface component input (in__pct)
- IP interface component output (out__pct)
- File system utilization (fs__pct)

  Modify the `aliases.dsv` file (located in the `/etc` directory under the ZenETL ZenPack) as needed until the changes are satisfactory when you re-run the script.

4. Once the output is satisfactory, re-run the script using the `--commit` option:

```
zendmd --script=../bin/chkaliases.zendmd [add | remove] --commit
```

On the /Server/Windows monitoring template, open the ProcessorTotalProcessorTime data point in the ProcessorTotalProcessorTime data source. You will see that an alias "cpu__pct" is defined with no RPN formula. In this case, the raw data point already represents the quantity we want ("Percentage of Consumed CPU Time"). The alias, cpu__pct, allows users to compare two different data sources.

The same principle applies to other data points. Some devices give network bandwidth by using kilobytes; others use megabits. Aliases provide a method to normalize these data points to a common denominator.

Analytics only extracts "aliases" from Resource Manager. A raw data point is never exported. To export a performance data point to Analytics, examine the data source/data point responsible for the raw value, and ensure that it has an alias that maps to an appropriate alias name whose value is properly normalized. Alias names are mapped to the meta_metric table in Analytics. (In previous versions, an alias would map to a fct_[alias] table; this mapping is now deprecated.) All performance data is now logically mapped to raw_v2_% and then to hourly_% and daily_% tables. These tables have a column named metric_key. The table meta_metric provides the lookup for a specific performance metric.

```
SELECT *
FROM raw_v2_cpu__pct
limit 5;
```

This query will provide all the performance data for the cpu__pct metric. The underlying data is stored in raw_%, hourly_% and daily_% tables. These tables should not be used directly, however if you decide to query against the raw_v2_% table, be aware that these tables are very big and you should be sure to include the device key, the component key, and the fact time stamp in your query whenever possible to fully leverage the available index. If the metric is for a device (no component included), the component key is 0.

With respect to event data points, all events are brought over to Analytics and are stored in the fct_event and fct_event_time tables.

Manually Modifying Data Points on the Resource Manager

To export data point data from a Resource Manager monitoring template, you can manually modify each data point.

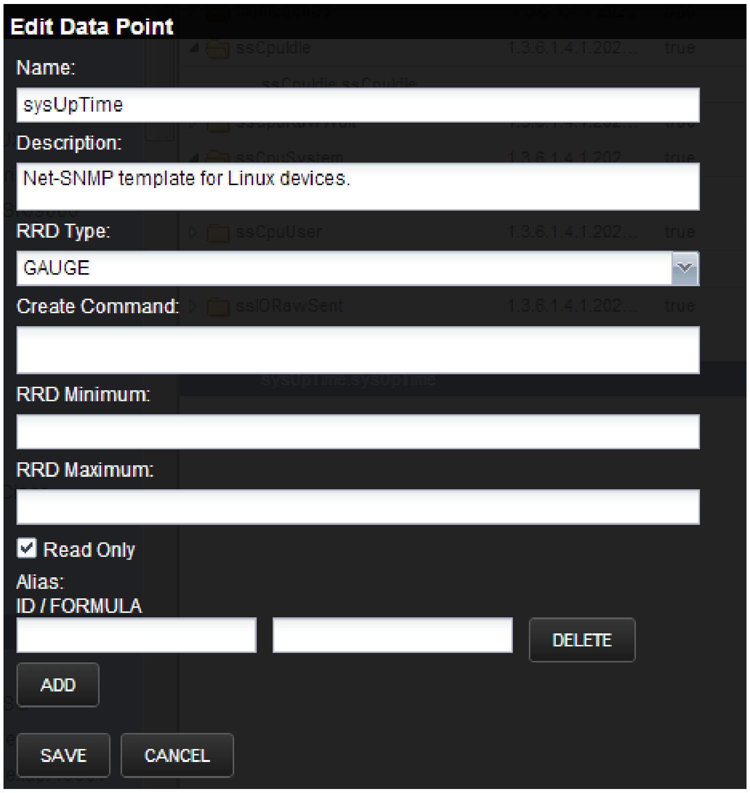
1. From the Resource Manager interface navigation bar, select **Advanced > Monitoring Templates > desired monitoring template**.

2. In the Data Sources area, expand the data source to see the associated data points.

3. Double-click the desired data point.

   The Edit Data Point dialog appears.

Figure 2.3. Edit Data Point


Edit Data Point

Name: sysUpTime

Description: Net-SNMP template for Linux devices.

RRD Type: GAUGE

Create Command:

RRD Minimum:

RRD Maximum:

☑ Read Only

Alias:
ID / FORMULA

4. Add an alias for the data point in the bottom section.

> ### Tip
>
> You can use the same name as the data point. No formula is required.

5. Click **Add**, and then click **Save**.

Once you have saved the data point with an alias, the data point will be included in the next extraction batch. You can verify that the data is being brought over by either of the following methods:

- View what data points are exported by viewing the "Meta-Data Management: Datapoint lineage" report provided with Analytics.
- Log in to the Analytics database directly and execute the following query:

  ```
  select distinct metric_name from reporting.meta_metric order by metric_name;
  ```

  The alias you created should appear in the returned list.

2.7.5. Configure SSL Communication

Zenoss Analytics fully supports the use of SSL for all of the different types of communication needed. It is assumed that you already have secured the Resource Manager behind SSL. This is a prerequisite for securing Analytics. For more information about securing Resource Manager, see *Zenoss Resource Manager Administration*.

The following four URLs must be provided to configure secure communication for the Analytics system. Following each of these URLs is a description of what the communication is used for:

- Internal Analytics server address
  - Resource Manager to call web service on the Analytics server to pull status of batches and scheduled batches
  - Zenmodeletl and zeneventetl daemons on Resource Manager to call web services on the Analytics server to work with batches and upload data files by HTTP put.
  - Zenperfetl daemon on each collector to call web services on the Analytics server to work with batches and upload data files by HTTP put.
- Internal Resource Manager server address
  - Analytics server to phone home to Resource Manager and authenticate and authorize a user.
- External Analytics server address
  - When the user logs into the Resource Manager web interface, the same login information is sent to the Analytics server. The user's password is passed in plain text if not using SSL.
  - When the user clicks on Reports > Advanced on the Resource Manager web interface, the user is redirected to the address specified.
- External Resource Manager server address
  - The Analytics server uses this to make links in reports that the user can click through to a device on the Resource Manager.

2.7.5.1. Install Apache and SSL on Analytics Server

The following procedure will install Apache and mod_ssl and set it to use a self-signed SSL certificate. You may choose to purchase a certificate signed by a third-party Certificate Authority or to generate your own certificate.

1. Log in to the Analytics server as the root user, or as a user with superuser privileges.

2. Install Apache.

   ```
   yum install httpd
   ```

3. Install Mod SSL for Apache.

   ```
   yum install mod_ssl
   ```

   Note

   If the installation of mod_ssl fails, check to see if there is an earlier version of open_ssl on the system that is being used by other packages. If this is the case, perform a `yum upgrade` to update the rest of the system and to bring in the correct version of open_ssl.

4. Add Apache to automatically start at the server start.

   ```
   chkconfig --levels 235 httpd on
   ```

5. Start Apache.

   ```
   service httpd start
   ```

6. Check your configuration by visiting both `http://yourserver.com/` and `https://yourserver.com/`.

2.7.5.1.1. Remove "Pragma" Headers

If you are supporting Internet Explorer 8 (IE8), you need to strip out the "Pragma" statements from the header of the HTTP files. Perform the following steps:

1. Log in to the Analytics server as the root user, or as a user with superuser privileges.

2. Navigate to the following Apache configuration folder.

   ```
   cd /etc/httpd/conf
   ```

3. Make a backup of the `httpd.conf` file.

   ```
   cp httpd.conf original_httpd.conf_original
   ```

4. Open `httpd.conf` in an editor.

   ```
   vi httpd.conf
   ```

5. Add the following line towards the top of the file.

   ```
   Header unset Pragma
   ```

6. Save the file and exit the editor.

7. Restart Apache.

   ```
   service httpd restart
   ```

2.7.5.2. Configure Apache Proxy and Rewrite

Perform the following steps to edit the `ssl.conf` file to add an internal proxy rule for Apache to proxy any request to the Analytics server and to turn on the Rewrite Engine:

1. Log in to the Analytics server as the root user, or as a user with superuser privileges.

2. Navigate to the following Apache configuration folder.

   ```
   cd /etc/httpd/conf.d
   ```

3. Make a backup of the `ssl.conf` file.

   ```
   cp ssl.conf original_ssl.conf_original
   ```

4. Open `ssl.conf` in an editor.

   ```
   vi ssl.conf
   ```

5. The last line of the file should be the closing tag `</VirtualHost>` which is the closing to the `<VirtualHost_default_:443>` tag. Add the following text just above the closing `</VirtualHost>` tag:

   ```
   #Internal proxy rule instructing Apache to proxy any request to the Analytics server on 7070
   ProxyPass /zenoss-analytics http://127.0.0.1:7070/zenoss-analytics
   ProxyPassReverse /zenoss-analytics http://127.0.0.1:7070/zenoss-analytics
   #Turn on the RewriteEngine
   RewriteEngine On
   #Redirect any just / over to /zenoss-analytics
   RewriteRule ^/+$ https://%{SERVER_NAME}:443/zenoss-analytics/
   ```

6. Save and close the `ssl.conf` file.

7. Restart Apache.

```
service httpd restart
```

2.7.5.3. Lock Tomcat to Localhost Only

The following steps should only be performed to lock down the Tomcat used by Analytics at the software level. An alternate solution is to simply close port 7070 altogether via firewall configuration.

1. Log in to the Analytics server as the root user, or as a user with superuser privileges.

2. Navigate to the following configuration folder.

   ```
   cd /opt/zenoss_analytics/conf
   ```

3. Make a backup of the `server.xml` file.

   ```
   cp server.xml original_server.xml_original
   ```

4. Open `server.xml` in an editor.

   ```
   vi server.xml
   ```

5. Locate the following section in the file.

   ```
   <Connector port="7070" protocol="HTTP/1.1"
   connectionTimeout="20000"
   redirectPort="8443"/>
   ```

6. Add in `address="127.0.0.1"` so that the section looks like the following:

   ```
   <Connector port="7070" address="127.0.0.1" protocol="HTTP/1.1"
   connectionTimeout="20000"
   redirectPort="8443"/>
   ```

7. Save and close the file.

8. Restart Tomcat to pick up the changes.

   ```
   service zenoss_analytics stop
   service zenoss_analytics start
   ```

2.7.5.4. Modifications to Resource Manager

Perform the following on the Resource Manager server to set the necessary HTTPS URLs:

1. Log in to the Resource Manager web interface as a user with Administrator privileges.

2. Navigate to Reports > Configuration.

   > **Note**
   >
   > If you have previously configured the 4 URLs for Analytics, you will see an error message at the top of the screen indicating that the system cannot connect to the Analytics server. This is expected behavior. Click ⚙ in the lower-left corner to open the configuration page again.

3. Modify the following 4 URLs with the values indicated:

   - Internal Analytics: HTTPS URL, including port (443), of the Analytics server.

   - Internal Zenoss: HTTPS URL, including port (443), of the Resource Manager server.

   - External Analytics: HTTPS URL, including port (443), of the Analytics server.
   - External Zenoss: HTTPS URL, including port (443), of the Resource Manager server.

4. Submit the form to save the settings.
5. Navigate to Advanced > Collectors.
6. Select each Hub and Collector one at a time and update them so that they receive the new settings.
7. Restart the zenmodeletl, zeneventetl, and zenperfetl daemons on the Resource Manager server so that they receive the new settings.

2.7.6. Configure Email Services on the Analytics Server

If you want to send the output of scheduled reports as email attachments, you must configure email services. To do this:

1. On the Analytics server, edit the `/opt/zenoss_analytics/webapps/zenoss-analytics/WEB-INF/js.quartz.properties` file.
2. Change the "host" and "from" values in the file.

   ```
   report.scheduler.mail.sender.host=localhost
   report.scheduler.mail.sender.username=
   report.scheduler.mail.sender.password=
   report.scheduler.mail.sender.from=zenoss@localhost
   report.scheduler.mail.sender.protocol=smtp
   report.scheduler.mail.sender.port=25
   ```

3. Save the file.
4. As the root user, restart Analytics for the changes to take effect:

   ```
   service zenoss_analytics restart
   ```

   #### Note

   Any configuration changes that you make below the `/opt/zenoss_analytics` folder will be overwritten during a subsequent upgrade. It is important to record the changes in some other directory so that these changes can be reapplied after an RPM upgrade. We recommend that you make a copy of these configuration changes directly in the `/opt/zenoss_analytics` folder.

2.8. Removing an Analytics Installation

To remove an installation of Analytics:

1. Log in to the Resource Manager master as the zenoss user, remove the ZenETL ZenPack from all masters:

   ```
   zenpack --remove ZenPacks.zenoss.ZenETL
   ```

2. Update all the collectors.
3. Log in to the Analytics server and change to the root user:

   ```
   su - root
   ```

4. Stop the Analytics server:

```
service zenoss_analytics stop
```

5. Remove the Analytics program files:

   ```
   rpm -e zenoss_analytics-version-build
   ```

   This command removes the program files, but does not remove configuration, log files, data files, or databases.

To remove configuration files, enter:

```
rm -rf /etc/zenoss_analytics
```

To remove data and log files, enter:

```
rm -rf /opt/zenoss_analytics
```

To remove databases (this will erase all your data):

1. Change to the zenoss user:

   ```
   su - zenoss
   ```

2. Enter the following command:

   ```
   mysql -u root -e 'drop database reporting; drop database zenoss_analytics';
   ```

Chapter 3. Upgrading to Analytics 4.3.1

This chapter provides instructions for upgrading Analytics version 4.3.0 to version 4.3.1. If you are upgrading from a version of Analytics prior to version 4.3.0, you must first upgrade to version 4.3.0 and then upgrade to version 4.3.1

> **Note**
>
> Perform an upgrade in a development or testing environment before performing it in a production environment. In particular, only Zenoss-maintained ZenPacks are tested and supported in an upgrade. Test all other ZenPacks for compatibility with this version of Analytics before upgrading a production environment.

The following matrix illustrates the available upgrade paths:

3.1. Prerequisite Tasks

Prior to upgrading to Analytics 4.3.1, perform the following tasks:

1. Download the following from the https://support.zenoss.com site:
    - RPM file for Zenoss DataStore, `zends-5.6.16-1.Version.elX.x86_64.rpm` or higher
    - RPM file for Analytics 4.3.1, `zenoss_analytics-4.3.1.0.0-1.Version.noarch.rpm`
    - EGG file for ZenETL ZenPack, `ZenPacks.zenoss.ZenETL-4.3.1.0.0.Version-py2.7.egg`
2. Wait for all batches in the staging state to complete.
3. As the zenoss user, stop the zenmodeletl and zeneventetl daemons on each attached Resource Manager server:

    ```
    zenmodeletl stop
    zeneventetl stop
    ```

4. As the zenoss user, stop the zenperfetl daemons on each attached remote collector and on the Resource Manager server:

    ```
    zenperfetl stop
    ```

3.2. Exporting the Reports Database

Zenoss Analytics uses JasperReports Server to run, display, schedule, and manage reports against the Analytics Data Warehouse. The JasperReports Server stores the report information in its own small database. This database needs to be exported as part of the upgrade process.

To export your existing reports database:

1. On the Analytics server as the `root` user, navigate to the jasperserver scripts folder:

    ```
    cd /opt/zenoss_analytics/jasperserver/buildomatic/
    ```

2. Run the export script:

    ```
    ./js-export.sh --everything --output-zip=/path_to_safe_place/backup.zip
    ```

3. Stop Analytics:

    ```
    service zenoss_analytics stop
    ```

3.3. Dropping the Reports Database

Once you have exported the reports database, perform the following to drop it:

1. Log in as the zenoss user:

   ```
   su - zenoss
   ```

2. Enter the following commands:

   ```
   zends -uroot
   drop database zenoss_analytics;
   ```

3.4. Upgrading Zenoss DataStore

The procedure in this section upgrades Zenoss DataStore on the Resource Manager master host. (If your installation includes a remote Zenoss DataStore host, perform the procedure in the next section, too.) The Resource Manager master host needs upgraded client libraries, both to communicate with Zenoss DataStore, and to copy the libraries to remote hub and collector hosts.

> ### Note
>
> The Zenoss DataStore upgrade installer preserves Resource Manager data, plus any customizations it finds, in the `/opt/zends/etc/zends.cnf` file.

1. Log in to the Analytics server as the root user, or as a user with superuser privileges.

2. If necessary, stop Zenoss DataStore.

   ```
   service zends stop
   ```

3. Upgrade Zenoss DataStore with the RPM you downloaded:

   ```
   RHEL/CentOS 5: yum -y --nogpgcheck localinstall zends-5.6.16-1.version.el5.x86_64.rpm
   RHEL/CentOS 6: yum -y --nogpgcheck localinstall zends-5.6.16-1.version.el6.x86_64.rpm
   ```

   You may see the following warning message:

   ```
   warning: /opt/zends/etc/zends.cnf created as /opt/zends/etc/zends.cnf.rpmnew
   ```

   If so, merge the new properties and values in `zends.cnf.rpmnew` with your customizations in `zends.cnf`.

   > ### Note
   >
   > Be sure to use version 5.6 of `zends`. Version 5.5 is not supported for Analytics 4.3.1.

4. Configure the Zenoss DataStore by adding the following lines to the end of the `/opt/zends/etc/zends.cnf` file:

   ```
   [mysqld]
   table_open_cache=16K
   table_definition_cache=16K
   tmp_table_size=2G
   max_heap_table_size=2G
   join_buffer_size=512K
   open_files_limit=200000
   ```

5. In the same `/opt/zends/etc/zends.cnf` file, edit the following lines to reflect the values listed below. The innodb_buffer_pool_size should be set to between 70 and 80 percent of total memory.

   ```
   innodb_adaptive_hash_index=OFF
   ```

```
innodb_buffer_pool_size=6G
innodb_log_file_size=1892M
innodb_log_buffer_size=128M
```

6. Enter the following commands:

   ```
   service zends start
   chkconfig --level 2345 zends on
   ```

   Then run `mysql_upgrade` since the zends minor version is changing from 5.5 to 5.6:

   ```
   mkdir /root/tmpdir
   /opt/zends/bin/mysql_upgrade -u root --tmpdir=/root/tmpdir
   ```

3.4.1. Upgrading a remote Zenoss DataStore (if applicable)

The procedure in this section upgrades a remote Zenoss DataStore. Perform this procedure *in addition to* upgrading Zenoss DataStore on the Analytics application server.

1. Log in to the remote Zenoss DataStore host as user `root`, or as a user with superuser privileges.

2. Stop Zenoss DataStore.

   ```
   service zends stop
   ```

3. Upgrade Zenoss DataStore with the RPM you downloaded:

   ```
   RHEL/CentOS 5: yum -y --nogpgcheck localinstall zends-version.el5.x86_64.rpm
   RHEL/CentOS 6: yum -y --nogpgcheck localinstall zends-version.el6.x86_64.rpm
   ```

   You may see the following warning message:

   ```
   warning: /opt/zends/etc/zends.cnf created as /opt/zends/etc/zends.cnf.rpmnew
   ```

   If so, merge the new properties and values in `zends.cnf.rpmnew` with your customizations in `zends.cnf`.

4. Start Zenoss DataStore.

   ```
   service zends start
   ```

5. Run `mysql_upgrade` since the zends minor version is changing from 5.5 to 5.6:

   ```
   mkdir /root/tmpdir
   /opt/zends/bin/mysql_upgrade -u root --tmpdir=/root/tmpdir
   ```

3.5. Upgrading the Analytics Server and Jaspersoft Database

Perform the following to upgrade the Analytics Server to version 4.3.1 and upgrade the Jaspersoft database:

1. Log in to the Analytics server as user `root`, or as a user with superuser privileges.

2. Using the Analytics RPM you downloaded, upgrade Analytics with one of the following commands:

   ```
   RHEL/CentOS 5: yum -y --nogpgcheck localinstall zenoss_analytics-version.noarch.rpm
   RHEL/CentOS 6: yum -y --nogpgcheck localinstall zenoss_analytics-version.noarch.rpm
   ```

   #### Warning

   Any configuration changes that you made below the `/opt/zenoss_analytics` folder will be overwritten during this upgrade. It is important to record the changes in some other directory so that these changes can be reapplied after this RPM upgrade. We recommend that you make a copy of these configuration changes directly in the `/opt/zenoss_analytics` folder, so that they can be restored after the upgrade.

3. Switch to the zenoss user:

   ```
   su - zenoss
   ```

4. Run the following commands:

   ```
   /opt/zenoss_analytics/bin/upgrade_db.py
   /opt/zenoss_analytics/bin/fix-theme.sh/path_to_safe_place/backup.zip
   /opt/zenoss_analytics/jasperserver/buildomatic/js-import.sh --input-zip=/path_to_safe_place/backup.zip
   ```

3.6. Importing the Reports Database

If you have created custom reports for a previous version of Analytics, contact Customer Support or Customer Services to migrate these reports to version 4.3.1, since there have been data warehouse schema changes.

3.7. Installing the ZenETL ZenPack on the Resource Manager

After installing the Analytics server, install the ZenETL ZenPack on the Resource Manager master. You can install it using the command line or from the Resource Manager interface. See the corresponding directions below:

To install using the command line:

1. As the zenoss user, enter the following command based on the version you are installing:

   ```
   zenpack --install ZenPacks.zenoss.ZenETL-Version-py2.7.egg
   ```

2. Create aliases to the data points on the monitoring templates that you want to bring into Analytics. See the "Running the chkaliases script" for instructions on using the script found in the ZenETL ZenPack.

3. Restart the system:

   a. Enter the following commands:

      ```
      zenoss stop
      ps -eaf | grep $ZENHOME
      ```

   b. For any process that is reported as running, perform a `kill -9` on those processes.

   c. Start the system:

      ```
      zenoss start
      ```

   ### Note

   Optionally, if you are using `daemons.txt` on the master, you should add entries for `zeneventetl`, `zenmodeletl`, and `zenperfetl` to the file. If you are using `collectordaemons.txt` on the collectors, you should add an entry for `zenperfetl` to the file.

When you restart the Resource Manager master, three additional daemons (`zeneventetl`, `zenmodeletl`, and `zenperfetl`) are started. The `zenperfetl` daemon is started on each remote collector.

Be sure to install the latest RPS for your Resource Manager instance so that you get all the latest fixes. The Distributed Collector ZenPack should be at version 2.5.12 or higher otherwise you will be required to use a `daemons.txt` and a `collectordaemons.txt` file to stop `zenmodeletl` and `zeneventetl` on running collectors.

3.8. Adding New Aliases

If you haven't already done so, refer to the "Export Data Point Data" section of the Installing and Configuring Analytics chapter to add new aliases to desired data points. You can create aliases using the chkaliases script or by manually adding them to a data point.

3.9. Starting the Analytics Service

Once the Zenoss system has been restarted after the upgrade, you will need to start the Analytics service:

1. Log in to the Analytics server as `root`, or as a user with superuser privileges.
2. Enable chkconfig for the Analytics service:

   ```
   chkconfig zenoss_analytics on
   ```

3. Start the Analytics service:

   ```
   service zenoss_analytics start
   ```

   ### Note

   Starting the Analytics service after an upgrade will take some time.

3.10. Post-Upgrade Steps

The upgrade from version 4.3.0 to 4.3.1 does not require any event or raw data migration, however there is some raw performance data that does need to be manually migrated. The Analytics service must be running and communicative before this migration takes place.

Perform the following to migrate raw performance data:

1. Log in to the Analytics server as the `zenoss` user.
2. Run the following migration call to migrate all the remaining performance data. If you want to limit your migration, skip to the next step.

   ```
   echo "call migrate_431_all_remaining();" | zends -u root reporting
   ```

3. If you want to limit the migration of performance data to a certain number of days, enter the following:

   ```
   echo "call migrate_431_remaining(X);" | zends -u root reporting
   ```

   where X is an integer representing the number of most recent days.

   ### Note

   There is nothing that need to be done to migrate aggregated data. That function is automatically handled when Analytics starts up.

Chapter 4. Working with Analytics

Read the following sections for more information about:

- Starting and stopping the Analytics server
- Logging in to Analytics
- Creating batches
- Searching the Analytics repository

4.1. Starting and Stopping the Analytics Server

Use one of the following commands if you need to stop, start, or restart the Analytics server (such as when troubleshooting problems). While the server is stopped, no data is extracted.

To start Analytics, log in to the Analytics server as the `root` user and enter the command:

```
service zenoss_analytics start
```

To stop and then start Analytics, log in to the Analytics server as the `root` user and enter the command:

```
service zenoss_analytics restart
```

> **Note**
>
> If you restart Analytics, there can be a delay of up to 60 seconds as Analytics and Resource Manager synchronize. During this time, single sign-on may not work correctly.

To stop Analytics, log in to the Analytics server as the `root` user and enter the command:

```
service zenoss_analytics stop
```

> **Note**
>
> While the Analytics server is stopped, no data is extracted.

4.2. Logging in to Analytics

To access Analytics, you can:

- Log in to the Resource Manager interface, using your Resource Manager user name and password (Single Sign On)
- Log in directly to Analytics, using your Resource Manager user name and password or as an Analytics superuser

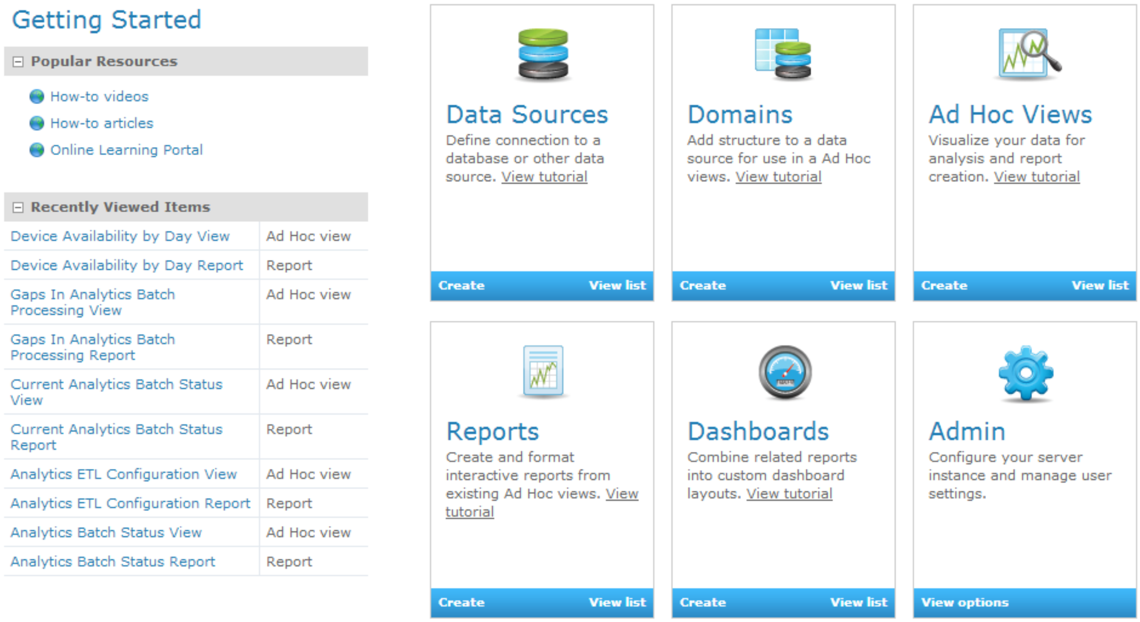
4.2.1. Logging in Through the Resource Manager Interface

Access to Analytics through single sign on (SSO) on the Resource Manager interface requires Manager, ZenManager, or ReportingUser privileges, assigned through roles. (Manager and ZenManager provide Analytics administrative privileges; ReportingUser provides user-level privileges.) You cannot log in to Analytics solely with Resource Manager administrator privileges. For more information on assigning roles, see the "Managing Users" chapter in *Resource Manager Administration*.

1. Log in to the Resource Manager interface with your Resource Manager user name and password.
2. Select Reports > Advanced.

The Getting Started page appears.

Figure 4.1. Getting Started



Getting Started

Popular Resources

- How-to videos
- How-to articles
- Online Learning Portal

Recently Viewed Items

Device Availability by Day View	Ad Hoc view
Device Availability by Day Report	Report
Gaps In Analytics Batch Processing View	Ad Hoc view
Gaps In Analytics Batch Processing Report	Report
Current Analytics Batch Status View	Ad Hoc view
Current Analytics Batch Status Report	Report
Analytics ETL Configuration View	Ad Hoc view
Analytics ETL Configuration Report	Report
Analytics Batch Status View	Ad Hoc view
Analytics Batch Status Report	Report

Data Sources

Define connection to a database or other data source. View tutorial

Domains

Add structure to a data source for use in a Ad Hoc views. View tutorial

Ad Hoc Views

Visualize your data for analysis and report creation. View tutorial

Reports

Create and format interactive reports from existing Ad Hoc views. View tutorial

Dashboards

Combine related reports into custom dashboard layouts. View tutorial

Admin

Configure your server instance and manage user settings.

From here, you can choose shortcuts to:

- View your reports
- Create a report
- Create an Ad Hoc View
- Change the Analytics server settings

Alternatively, make a selection from one of the Analytics menu options:

- **View** - View search results, the Analytics repository, or messages.
- **Manage** - Organizations, users, roles, or server settings (accessible by administrators)
- **Create** - Create ad hoc reports, dashboards, or domains.

Note

If you log out of Analytics after logging in through the Resource Manager interface, you must log out of Resource Manager to log in again through Resource Manager. Furthermore, if the Resource Manager server is restarted, you must log back in to Analytics.

4.2.2. Logging in Directly to Analytics

You can log in directly to Analytics by using your Resource Manager user name and password (with appropriate Resource Manager privileges), or as an Analytics superuser.

1. Browse to the URL of your Analytics server.

The Analytics login page appears.

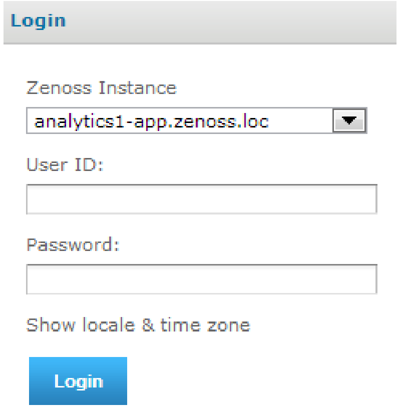
2. Log in:

- With your Resource Manager user name and password:

  a. From the Resource Manager Instance list of options, select the Resource Manager instance for which you will provide a user name and password.

  b. Enter your Resource Manager user name and password in the User ID and Password fields.

Figure 4.2. Analytics Login - Resource Manager Credentials



Zenoss Instance: analytics1-app.zenoss.loc

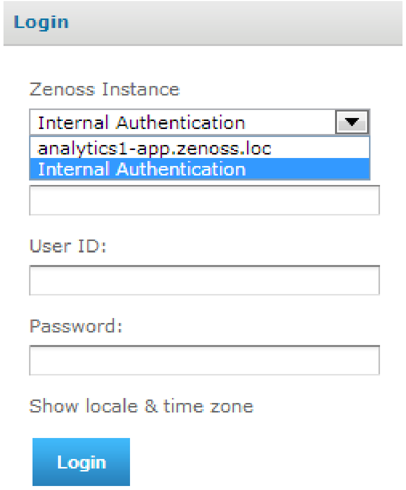
- As an Analytics superuser:

  a. From the Resource Manager Instance list of options, select Internal Authentication.

  b. If the user is located in an Analytics organization other than Resource Manager, enter the organization name. (Typically, you would not enter a value in this field.)

  c. Enter your Analytics User ID and password in the User ID and Password fields.

Figure 4.3. Analytics Login - Analytics Superuser



Zenoss Instance: Internal Authentication

3. Click **Login**.

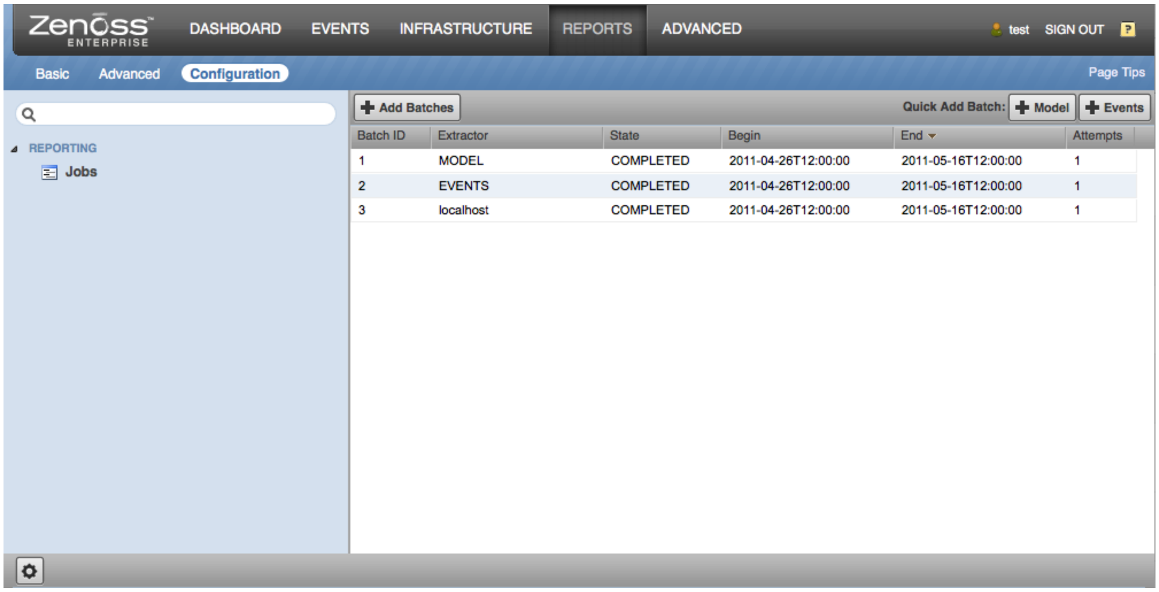
4.3. Creating Batches

You create batches to populate the reporting database. Every eight hours, the ETL processes a batch and extracts information from the previous eight-hour window. Three main types of batches match the primary data types (Performance, Model, and Event).

Existing batches and associated status information appear in the batch list. To access the batch list, select Reports > Configuration from the Resource Manager interface.

The batch list appears.

Figure 4.4. Batch List



Batch ID	Extractor	State	Begin	End	Attempts
1	MODEL	COMPLETED	2011-04-26T12:00:00	2011-05-16T12:00:00	1
2	EVENTS	COMPLETED	2011-04-26T12:00:00	2011-05-16T12:00:00	1
3	localhost	COMPLETED	2011-04-26T12:00:00	2011-05-16T12:00:00	1

The list shows each batch and its status (UNSTARTED, EXTRACTING, STAGING, FAILED, or COMPLETED).

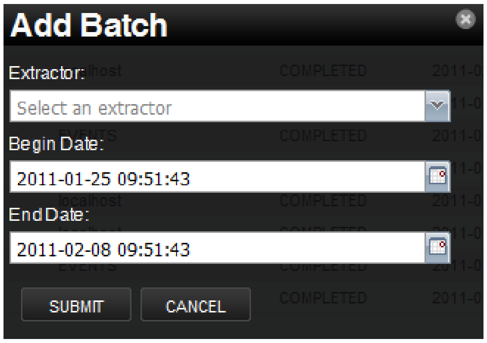
In general, you do not need to create manual batches. The only reason a manual batch should be created is as a result of a previous failed scheduled batch.

To add a manual batch:

1. From the batch list, click ✚ Add Batches (Add Batches).

   The Add Batch dialog appears.

Figure 4.5. Add Batch



Add Batch

Extractor:

Select an extractor

Begin Date:

2011-01-25 09:51:43

End Date:

2011-02-08 09:51:43

2. Make selections in the dialog:

   - **Extractor** - Select MODEL, EVENTS, or localhost. An additional performance extractor is listed for each remote collector. Extractors only appear once they phone home and self-register. This may take a few minutes after you connect the Resource Manager to the Analytics server and update the hubs and collectors.
   - **Begin Date** - Optionally adjust the begin date and time. By default, the value is set to two weeks before the current Analytics server date and time.
   - **End Date** - Optionally adjust the end date and time. By default, the value is set to the current Analytics server date and time.

   #### Note

   Creating a batch that finishes in the future will leave a batch with a status of UNSTARTED. It will not start until that future date is reached.

   #### Note

   Setting dates for MODEL is ignored, you will just get the current model.

3. Click **Submit**. The batch is added to the batch list.

Note

RRD stores data samples at various different resolutions. It is configured to store a fixed number of data samples at any resolution. Resource Manager, by default, creates RRD files to store 288 samples at five minute (300 second) intervals, called a *step*. This means that Resource Manager retains, at most, one day of five-minute samples for a data point. To ensure that disk usage does not grow without bound, RRD consolidates the five-minute samples in less granular periods (such as 15 minutes, two hours, one day, or one week).

When you schedule batches, remember that Resource Manager will return data at the finest granularity that exists over the entire period. This means that, for example, if your query is for the last 26 hours, you will not receive 300 second data samples but approximately 1800 second (15 minute) samples. The further back your batch periods reach, the less granular the retrieved data will be.

4.3.1. Configuring Repeat Intervals

When configuring batch intervals (in Resource Manager), you must set an Analytics repeat interval for each Analytics server (rather than for each Resource Manager instance). This means that the same Analytics repeat interval setting will appear across Resource Manager instances.

You can configure repeat intervals separately for each extractor type. By default, all repeat intervals are set to a value of 8 hours.

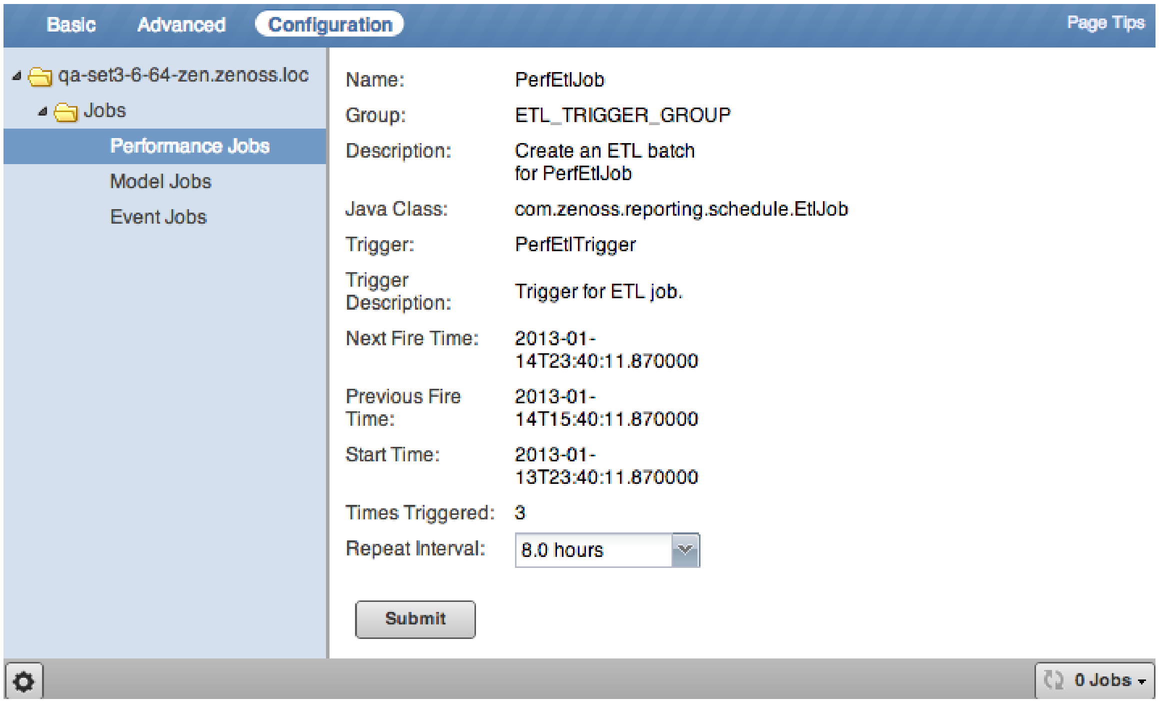
Note

If you upgraded to Analytics 4.3.1, you will need to edit the repeat interval to the value you want, since the upgrade procedure sets all repeat intervals back to the default value of 8 hours.

To edit the repeat interval:

1. Expand the Jobs folder, and then select one of the extractor types.

Figure 4.6. Edit Repeat Interval



Name: PerfEtlJob

Group: ETL_TRIGGER_GROUP

Description: Create an ETL batch for PerfEtlJob

Java Class: com.zenoss.reporting.schedule.EtlJob

Trigger: PerfEtlTrigger

Trigger Description: Trigger for ETL job.

Next Fire Time: 2013-01-14T23:40:11.870000

Previous Fire Time: 2013-01-14T15:40:11.870000

Start Time: 2013-01-13T23:40:11.870000

Times Triggered: 3

Repeat Interval: 8.0 hours

2. Select a repeat interval (.25 hours to 8 hours), and then click **Submit**.

4.3.2. Setting a Delay on Performance Batches

A configuration option for the `zenperfetl` daemon allows you to delay the execution of a performance batch until it reaches a specified amount of time into the past. This defaults to fifteen minutes and should not be changed without consulting Zenoss Customer Support.

- **perfbatchdelayinminutes** - Run only those performance batches with an end time at least *x* minutes in the past.

To set a value for this option, select Advanced > Daemons, and then click the option to edit the daemon configuration.

Note

Performance and event data is loaded as quickly as possible on the Analytics server, however, staged model data is only looked for once every 4 hours, by default.

4.4. Configuring the Analytics Data Retention Policy

Based on your business needs, you may want to change the data retention policy for certain types of data. The data retention policies are configured through direct manipulation of the `meta_setting` table in the data warehouse ("reporting") database.

To see the data retention setting in the `meta_setting` table, perform the following:

1. Log in to the Analytics data warehouse database as the `root` user or a user with administrative permissions.

   ```
   zends -u root reporting
   ```

2. Enter the following command at the zends prompt:

   ```
   zends> select * from meta_setting where setting_name in
   ('daily_retention_days','raw_retention_days','event_retention_days','hourly_retention_days');
   ```

   A table similar to the following is displayed.

setting_key	setting_name	setting_value	setting_description
1	daily_retention_days	365	Days to keep daily aggregation data.
2	raw_retention_days	45	Days to keep raw fact data.
3	event_retention_days	365	Days to keep event data.
10	hourly_retention_days	90	Days to keep hourly aggregate data.

The value displayed in the `setting_value` column specifies the number of days for which the data is kept for the given type of data. Raw, Hourly, and Daily retention refer to performance data only. Raw indicates the lowest resolution of performance data extracted from the RRD files (typically 5 minute data) whereas Hourly and Daily represent hourly and daily aggregates of that data. The values shown in the above table are the Zenoss recommendations, which are satisfactory for typical reporting and analysis needs. The Event retention applies to all events extracted to Zenoss Analytics.

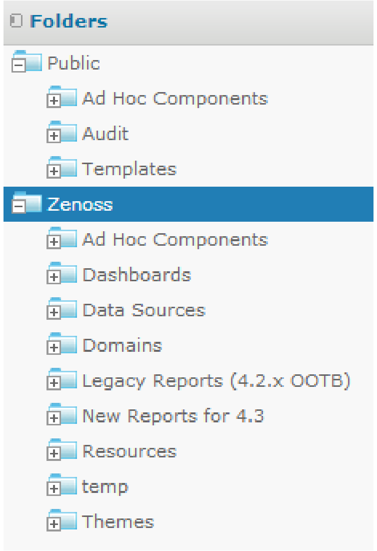
Data is automatically purged by the system on a daily basis (typically runs at midnight) based on these settings. Updating them takes effect immediately at the next data purge. To change one of these settings, update the value in the table. For example, to change the retention policy for raw performance data from 45 days to 30 days, issue the following database update at the zends prompt:

```
zends> update meta_setting set setting_value = '30' where setting_name = 'raw_retention_days';
```

4.5. Working with the Analytics Repository

The repository is Analytics' internal storage for all reports, dashboards, analysis views, and related files. The repository is organized as a tree structure of folders containing those components, much like a file system. Zenoss provides a set of folders that are provided upon installation (ad hoc components, dashboards, domains, reports, resources, etc.) as shown below. To access the Repository Folder view, select View > Repository.

Figure 4.7. Zenoss Folder in Repository



Folders

- Public
  - Ad Hoc Components
  - Audit
  - Templates
- Zenoss
  - Ad Hoc Components
  - Dashboards
  - Data Sources
  - Domains
  - Legacy Reports (4.2.x OOTB)
  - New Reports for 4.3
  - Resources
  - temp
  - Themes

Since the offerings in the Zenoss-provided folders may change in a future update, it is important to note that any changes you make to the Zenoss folder may be overwritten in the future. Therefore, it is important to populate a folder specifically for your organization. You can use the contents of the Zenoss folder as a guide to determine which folders you would like to copy to your organization's space in the repository. To copy a folder, simply right-click on the folder and select Copy, then right-click on the destination parent folder and select Paste.

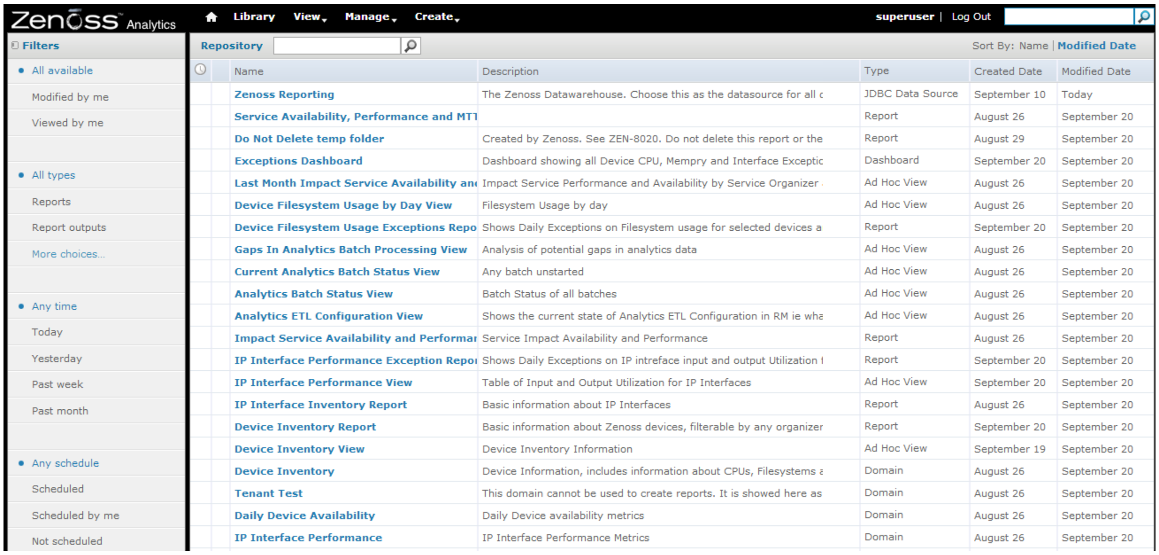
Note

Analytics stores all objects and files in a private database that is otherwise inaccessible, except to administrators.

4.5.1. Searching the Repository

Once you have set up your organization's folder within the repository, you can easily search for your information for quick access. To search the repository, select View > Search Results.

Figure 4.8. Analytics Repository Search Results



Name	Description	Type	Created Date	Modified Date
Zenoss Reporting	The Zenoss Datawarehouse. Choose this as the datasource for all c	JDBC Data Source	September 10	Today
Service Availability, Performance and MTT		Report	August 26	September 20
Do Not Delete temp folder	Created by Zenoss. See ZEN-8020. Do not delete this report or the	Report	August 29	September 20
Exceptions Dashboard	Dashboard showing all Device CPU, Mempry and Interface Exceptic	Dashboard	September 20	September 20
Last Month Impact Service Availability an	Impact Service Performance and Availability by Service Organizer	Ad Hoc View	August 26	September 20
Device Filesystem Usage by Day View	Filesystem Usage by day	Ad Hoc View	August 26	September 20
Device Filesystem Usage Exceptions Repo	Shows Daily Exceptions on Filesystem usage for selected devices a	Report	September 20	September 20
Gaps In Analytics Batch Processing View	Analysis of potential gaps in analytics data	Ad Hoc View	August 26	September 20
Current Analytics Batch Status View	Any batch unstarted	Ad Hoc View	August 26	September 20
Analytics Batch Status View	Batch Status of all batches	Ad Hoc View	August 26	September 20
Analytics ETL Configuration View	Shows the current state of Analytics ETL Configuration in RM ie wha	Ad Hoc View	August 26	September 20
Impact Service Availability and Performar	Service Impact Availability and Performance	Report	August 26	September 20
IP Interface Performance Exception Repo	Shows Daily Exceptions on IP intreface input and output Utilization f	Report	September 20	September 20
IP Interface Performance View	Table of Input and Output Utilization for IP Interfaces	Ad Hoc View	September 20	September 20
IP Interface Inventory Report	Basic information about IP Interfaces	Report	August 26	September 20
Device Inventory Report	Basic information about Zenoss devices, filterable by any organizer	Report	September 20	September 20
Device Inventory View	Device Inventory Information	Ad Hoc View	September 19	September 20
Device Inventory	Device Information, includes information about CPUs, Filesystems a	Domain	August 26	September 20
Tenant Test	This domain cannot be used to create reports. It is showed here as	Domain	August 26	September 20
Daily Device Availability	Daily Device availability metrics	Domain	August 26	September 20
IP Interface Performance	IP Interface Performance Metrics	Domain	August 26	September 20

The Repository Search Results page displays search results, as well as lists of reports and ad hoc views. It features:

- Search controls - Find resources by searching for text in their names or descriptions.
- Filters for refining results - Provides a quick way to find resources based on other criteria, such as type or access time.
- Search expression and sorting - Interactive status bar that shows search criteria, filters, and sorting criteria.
- List of resources - Resources in the repository that meet the search criteria. Click or right-click a resource in the list to view it or see available actions.

Repository Search Results page allows you to refine or enlarge your set of results. Analytics remembers your settings so that your most commonly needed resources remain visible when you return to the page.

The following sections provide more information about:

- Searching by name
- Refining with filters
- Changing the search expression
- Interacting with the list of resources

4.5.1.1. Searching by Name

Use the search field at the top of any Analytics page to begin a search. Search terms are not case sensitive. When you click the search icon, Analytics opens the repository page with the search term and displays the results.

Analytics matches the search term, even word fragments, anywhere it appears in the display name or description string of the results. Analytics does not match folder names or display folders in the results. If you enter multiple words, Analytics searches for resources that match all words, but not necessarily in order. Do not enter quotes around terms or any symbols between terms.

If you enter a search term and click Search on the Repository page, the search uses the current settings on the page. If you enter a search term and click the search icon at the top of any other page, the search uses the default settings:

- Include subfolders
- Start at the top-most visible folder
- Sort alphabetically by name

4.5.1.2. Refining with Filters

The filters below the folder tree let you refine your search using several other resource attributes. For example, these filters can help you find your most recently viewed reports. You can set each filter independently of the others.

The user filter has the following settings:

Filter Setting	Description
All available (default)	Does not take into account the user's access to the resource.
Modified by me	Selects only resources that have been last modified by the currently logged in user.
Viewed by me	Selects only resources that have been run and viewed by the currently logged in user. This filter applies not only to visualization types, but also to resources that are included in reports (such as images).

The resource type filter has the following settings:

Filter Setting	Description
All types (default)	Does not take into account the resource type; as a result, every type is displayed.
Reports	Displays only reports.
Report outputs	Displays only the output from reports that were scheduled or run in the background. Report output can be any of the supported export types (such as HTML or PDF).
Ad Hoc views	Displays only the output from reports that were created using an ad hoc view into an available data source type (Topics, Domains, or OLAP Connections).
Dashboards	Displays only dashboards.
OLAP views	Displays only OLAP views.
Domains	Displays only domains.
Data sources	Displays only data sources.
Deprecated Reports	Displays only deprecated reports.

The access time filter has the following settings. All times are relative to your effective time zone:

Filter Setting	Description
Any time (default)	Does not take into account the last modification time of a resource.
Today	Resources viewed or modified since the previous midnight.

Filter Setting	Description
Yesterday	Resources viewed or modified during the previous day ending at midnight.
Past week	Resources viewed or modified during the past 7 days, including today.
Past month	Resources viewed or modified during the past 30 days, including today.

The scheduled report filter has the following settings:

Filter Setting	Description
Any schedule (default)	Does not take into account the existence of scheduled jobs.
Scheduled	Only reports that have scheduled jobs.
Scheduled by me	Only reports that have jobs scheduled by the currently logged in user.
Not scheduled	Only reports that do not have scheduled jobs, and all other resource types.

4.5.1.3. Changing the Search Expression

Above the list of resources, the search expression shows you all of the criteria that contribute to the search. It contains the following elements, always in this order from left to right:

- The search term, if any, or the word All
- Any and all filter settings

The search expression provides a quick visual summary of the search criteria for the list of results that appear right below it. The search expression is also interactive, allowing you to easily remove some of the filters. It supports the following actions:

- If there is more than one filter, clicking any of them removes all those to its right.
- You can click the search term or the word All to remove any folders to the right.

Note

To remove the search term, click the X icon in the search field. The search expression then displays All.

After any of these actions, the search controls displayed on the left are refreshed, and the results are updated.

To the right of the search expression, the sort criteria lets you change the order of the results. Analytics supports the following sorting:

- Click **Name** to sort alphabetically from A-Z. This is the default sort order.
- Click **Modified Date** to sort by the latest modified time and date (most recent at top).

4.5.1.4. Interacting with the List of Resources

Once the resource or resources you want are displayed, you can interact with them in several ways:

- Click the name of a report or dashboard to run and view it.

- Right-click the name of a resource to access other operations on the context menu, for example Open in Designer... Items appear on the context menu according to your permissions.
- Select the resource name or click anywhere in the row to select it. (Control-click anywhere in the rows to select multiple resources.) You can drag and drop selected items to move them or press Control while you drag and drop items to copy them.

Two icons may appear between a report name and its selection box:

- + indicates that the report has saved options for is input controls. Click + to list the saved options below the report.
- The clock icon indicates that the report has been scheduled to run or that it currently is running in the background. Click this icon to view the list of jobs scheduled for the report.

4.5.1.5. Workflow for Creating Your Own Reports

The following sections describe the process to follow for creating your own domains, then creating ad hoc views based on those domains, and finally creating and scheduling reports to be run from those ad hoc views.

4.5.2. Creating Domains

A domain is a metadata layer that provides a business view of the data accessed through a data source. A domain presents the data in business terms appropriate to your audience, and can limit the access to data based on the security permissions of the person running the report. A domain can be used to create reports and ad hoc views. Zenoss provides several out-of-the-box domains, but you may want to create your own domains based on business needs. This section discusses some of the considerations to make note of as you create a domain. For more detailed information about creating domains, see the *JasperReports Server User Guide*.

In practice, you should plan domain features before you begin the creation process. You will use the `zenoss_reporting` data source for your domain creation. Decide on the element of your design: tables and columns to choose, joins to perform, filters to define, sets you need, and item properties to expose to users. Finally, determine a security policy for the data retrieved through the domain.

To help you in your creation of domains, we have included an Entity-Relationship Diagram (ERD) in Appendix A that shows you the relationships between tables in the database. You will also find a sample of some dynamically created dimension tables as well as a flow of performance fact data as it goes through hourly and daily aggregation.

To create a new domain:

1. Log in to Analytics as an administrator or superuser.
2. Click Create > Domain. The Add New Domain screen is displayed.
3. Fill in the name, resource ID and a description.
4. Select a Save Location by browsing to your organization's Domain folder.
5. In the Data Source field, browse to the `zenoss_reporting` data source located at `/organizations/zenoss/Data_Sources/zenoss_reporting`
6. In the Domain Design section, click the Edit with Domain Designer link. The Domain Designer screen is displayed. On this screen you will configure various aspects of the design. Just click the corresponding tab for the following:
    - Tables - Select all the tables whose columns you want to use in the domain, either directly or indirectly.
    - Derived Tables - Enter queries whose results appear as derived tables in the domain.
    - Joins - Define inner and outer joins between all the tables and derived tables.

- Calculated Fields - Enter expressions whose results appear as calculated fields. An example of a calculated field can be found in the IP Interface Inventory domain found in the `Zenoss/Domains` folder. Note that all the `_deleted` calculated fields are based on whether the time stamp is present or not.

- Pre-filters - Specify conditions on field values to limit the data accessed through the domain. An example of a pre-filter can be found in the Daily Device KPI domain found in the `Zenoss/Domains` folder. You cannot use this domain if you have components. Thus, there is a pre-filter defined as `component_key = 0`.

- Display - Organize the visual aspects of the domain and change the display properties of tables, columns, sets, and items exposed to domain users. The properties are located in the far-right column. The Label is the friendly text display, while the ID is the system name that shouldn't be changed unless there are similar IDs coming from different input controls. In the Summary Function section, all performance metrics are defined as measures, while everything else in the system (including time) is defined as a field.

7. As you create your domain, click the Check Design button to validate your domain design. If you have issues, click the Export Design button which will create an XML file of your schema that can be shared with Zenoss Support.

8. Click OK once you have finished with the Domain Designer. You will return to the Edit Domain screen.

9. If all the information is correct, including your design, click Submit to add the domain to the repository.

4.5.3. Creating Ad Hoc Views

The Ad Hoc editor supports the creation of views for various types of reports: tables, crosstabs, and charts. You interact with the editor to create these views by dragging and dropping elements. You can add and summarize fields, define groups, label and title the report, and format data for each field.

To open the Ad Hoc editor:

1. Log in to Analytics as an administrator or superuser.

2. Click Create > Ad Hoc View. The Data Chooser: Source screen is displayed.

3. Select your data source:

   - Topics - These can be either a JRXML file that is associated with a data source on the server or can be created from a domain with one or more filters applied to it. Using a topic as your source generates an empty view, which allows you to begin adding data to your view right away, without choosing, pre-filtering, or changing display names of the data (all of which are required steps when creating a Domain-based view.

   - Domains - They specify tables in the database, join clauses, calculated fields, display names, and default properties, all of which define items and sets of items for creating Ad Hoc views.

   - OLAP Connections - These expose transactional data and define how the data can be seen as a multidimensional cube. Note that you can only create chart and crosstab views from an OLAP connection.

4. Select a view type:

   - Tables - Used to view values in the database and to summarize the values in columns.

   - Charts - Used to compare one or more measures across multiple sets of related fields. Charts summarize data graphically. Some types of charts include bar chart, line chart, and pie chart, among others.

   - Crosstabs - Used to aggregate data across multiple dimensions. Crosstabs are more compact representations than tables; they show only computed values, rather than individual database values. Columns and rows specify the dimensions for grouping; cells contain the summarized measurements.

5. Fill out the New Ad Hoc View screen based on the view type you selected. For more details on the functionality and toolbar icons, see the *JasperReports Server User Guide*.

6. When completed, save your Ad Hoc View. This view is saved in the repository. You can use this view to create reports or you can save the view as a report itself so that you can embed the data content into a dashboard or see the data in the interactive report viewer.

Note

When you create a report from an Ad Hoc view, the report is considered "dependent" on that view. If you later edit the Ad Hoc view, the dependent reports are not updated with the new changes to the view.

4.5.4. Creating and Running Reports

Zenoss Analytics makes it easy to create and run reports. When you run a report, it opens in the interactive Report Viewer. Using this Viewer, you can personalize and refine the displayed report data. If your report has input controls, you can run the report with one set of data and then another. You can then export the output to another format or you can schedule the report to create an output repeatedly and unattended during off hours or at other times.

To create a report and edit the output:

1. Log in to Analytics.

2. From the Home page, click Create Report; or from any page, click Create > Report.

3. Select an Ad Hoc view from the Create Report window, then click OK.

4. If Input Controls are asked for, provide them in the displayed window. If not, the report output appears.

5. Edit the report output by clicking the Options button in the upper-right corner.

Note

By default, a dynamic date range is used to define the date filter. The range is defined by the unit (DAY, WEEK, MONTH) and a numerical unit that can be positive or negative. For example a range of DAY-7 to DAY is equivalent to the last seven days from when the report is run.

6. While you cannot change the fundamental type of output (you cannot change a chart into a crosstab), you can change various elements of the output based on the type of report. For example, you can resize columns or filter by a column's value. If you are using a chart, you can change the type of chart you want to display. Refer to the *JasperReports Server User Guide* for even more ways to customize your reports.

7. Once you have the report output displaying the way you want, save the report by clicking the Save icon.

To schedule a report job to be run:

1. Log in to Analytics.

2. On the Home page, click View Reports; or from any page, click View > Repository.

3. Use the search field or browse the list of reports to find the report you want to schedule.

4. Right-click the report name and select Schedule from the context menu. The Schedule Jobs page appears.

5. Click Schedule Job. The Set Up the Job wizard appears. Fill out the information requested and select the recurrence of the job. The default value is None, which will run the report only once. Click Simple to schedule the job to recur at an hourly, daily, or weekly interval or click Calendar to schedule the job to recur on certain days of the week or days of the month. Click Next to continue the wizard.

6. Set the Parameter Values and click Next to schedule the report. On the Repository view page, you will see a clock icon to indicate that a report has been scheduled to run.

Note

The permissions of the user who schedules a job determines the data that the report exposes. Be aware that sensitive data could be exposed if you schedule a job as an administrative user with no data restrictions because the report output will contain all the requested data in the data source. Any user who receives the report output can view all the data regardless of the user's access restrictions.

4.5.5. Entity-Relationship Diagrams for Aggregation-related Tables

The following Entity-Relationship Diagrams (ERD) show the relationships between aggregation-related tables in the database.

Figure 4.9. Configuration ERD of Aggregation-related table

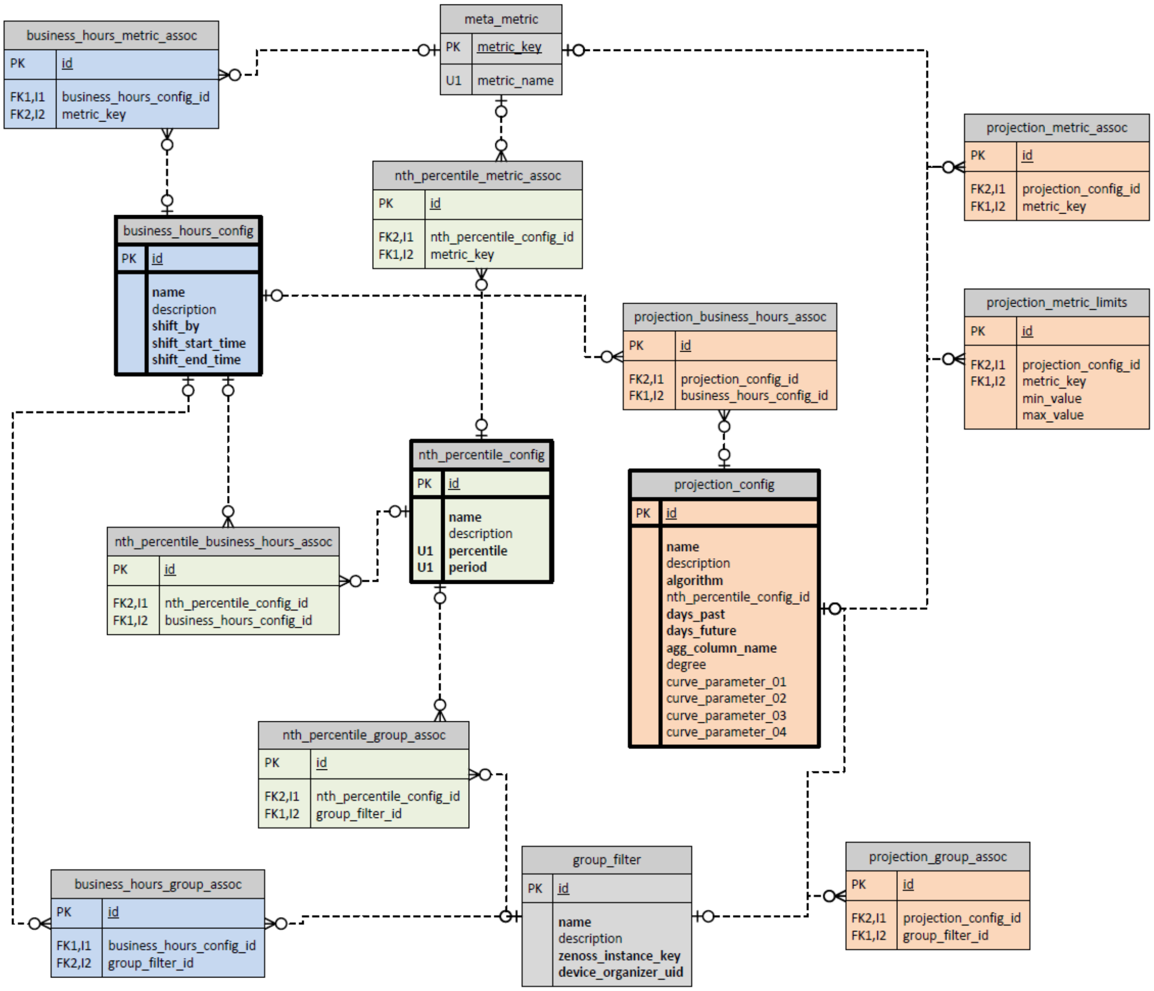
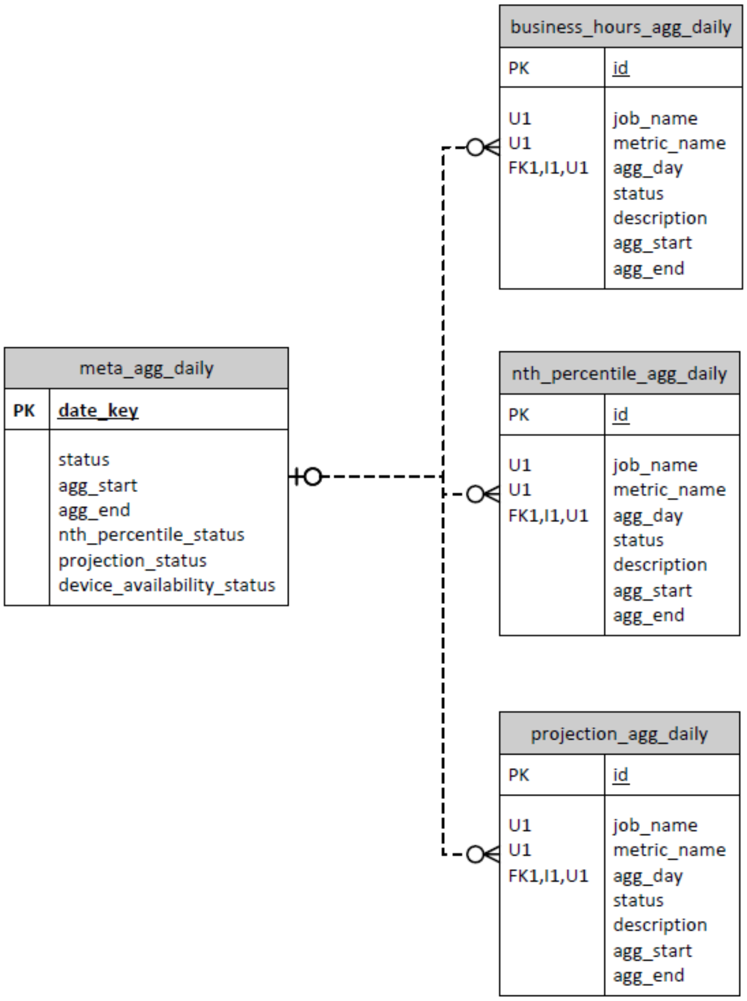


Figure 4.10. Status Tracking ERD of the Aggregation Execution



4.5.6. Creating Business Hours Aggregation

You can prepare reports showing information relative to the business hours of operation based on a user-specified time offset, which allows for more accurate reporting of how applications and services are performing relative to different user communities. For example, you can aggregate data for a specified time range that has been defined as a "business day" or "shift". Each row is named for the shift or business hours it represents, e.g., `austin`, `london`, `singapore` as locations or shifts like `austin_day`, `austin_swing`, `austin_night`.

The creation of rows for the business hour aggregation is performed manually by inserting records into the reporting database. The tables of interest are described below:

- `business_hours_config`: main table defining the business hours
  - `name`: name of the shift or business hours
  - `description`: up to 255 characters to describe the row (optional)
  - `shift_by`: difference in time between the raw data time stamp and the time specified by the `shift_start` and `shift_end` fields. This value can be negative and is in the format `hh:mm`.
  - `shift_start_time`: time of day when the shift (or business day) begins. The format is `hh:mm`.

  - `shift_end_time`: time of day when the shift (or business day) ends. This value can be less than `shift_start` which indicates that the end of the shift is the following day. The shift begins at the `shift_start` time and goes up to but does not include the `shift_end` time. The format is `hh:mm`.

shift_start_time	shift_end_time	shift_by	Selected range of raw data
08:00	16:00	04:00	12:00 PM through 8:00 PM of the current day
17:00	02:00	00:00	Assuming today is 5/15/2013: 5/15/2013 5:00 PM through 5/16/2013 2:00 AM
06:00	14:00	-11:30	Assuming today is 5/15/2013: 5/14/2013 6:30 PM through 5/15/2013 2:30 AM

- `group_filter`: optional filter used to reduce the set of devices included in the aggregation. If a location is used, such as London, you may only want to include devices in the London data center. This filter uses a `zenoss_instance_key` from `dim_group`, so groups need to be created and populated using the grouping procedure from the user interface. If this filter is not provided for a business hour row (see `business_hours_group_assoc` below), all devices are included in the aggregation.
  - `filter_name`: name to identify the filter
  - `description`: up to 255 characters to describe the filter (optional)
  - `zenoss_instance_key`: key of the zenoss instance, used along with `device_organizer_uid` to identify a device grouping in a particular zenoss instance
  - `device_organizer_uid`: id of the device group, used along with `zenoss_instance_key` to identify a device grouping in a particular zenoss instance
- `meta_metric`: optional filter used to reduce the set of metrics included in the aggregation. If this filter is not provided for a business hour row (see `business_hours_group_assoc` below), all metrics are included in the aggregation.
  - `metric_key`: primary key
  - `metric_name`: name of the metric to aggregate
- `business_hours_group_assoc`: each business hour row may have multiple associated job filters. Each job filter may be used by many business hour rows. For each job filter associated with a business hour row, an association record is created.
  - `business_hours_config_id`: id of the row in the business hours table
  - `group_filter_id`: id of the job filter
- `business_hours_metric_assoc`: each business hour row may have multiple associated metric filters. Each metric filter may be used by many business hour rows. For each metric filter associated with a business hour row, an association record is created.
  - `business_hours_config_id`: id of the row in the business hours table
  - `metric_key`: id of the metric filter

4.5.6.1. Inserting Records for Business Hours Aggregation

Perform the following steps to insert records into the reporting database. The code in this procedure is for demonstration purposes only. This example assumes that you have two data centers (Austin and Bangalore) already defined and the metrics you want to filter on are called `analytics_test1` and `analytics_test2`:

1. Log in to the Analytics reporting database as the `root` user or a user with administrative permissions.

   ```
   zends -u root reporting
   ```

2. Enter the following commands at the zends prompt substituting values that apply to your company's situation. The first section of the following inserts groups. If you already have groups, ignore this section and customize the commands to your situation:

   ```
   zends> insert into dim_group
      (group_key, sid, zenoss_instance_key, id, device_organizer_name, device_key,
          device_organizer_uid)
   values
      (1001, '8525e4e2-4da4-4da5-8d4c-20ec70eaae0b/analytics_test_1', 1, 'AustinDataCenter',
         'AustinDataCenter', 1, '/zport/dmd/Groups/AustinDataCenter'),
      (1002, '8525e4e2-4da4-4da5-8d4c-20ec70eaae0b/analytics_test_3', 1, 'AustinDataCenter',
         'AustinDataCenter', 3, '/zport/dmd/Groups/AustinDataCenter'),
      (1003, '8525e4e2-4da4-4da5-8d4c-20ec70eaae0b/analytics_test_2', 1, 'BangaloreDataCenter',
         'BangaloreDataCenter', 2, '/zport/dmd/Groups/BangaloreDataCenter'),
      (1004, '8525e4e2-4da4-4da5-8d4c-20ec70eaae0b/analytics_test_4', 1, 'BangaloreDataCenter',
         'BangaloreDataCenter', 4, '/zport/dmd/Groups/BangaloreDataCenter');

   insert into group_filter
      (name, description, zenoss_instance_key, device_organizer_uid)
   values
      ('austin', 'Devices in the Austin data center', 1, '/zport/dmd/Groups/AustinDataCenter'),
      ('bangalore', 'Devices in the Bangalore data center', 1,
          '/zport/dmd/Groups/BangaloreDataCenter');


   insert into meta_metric (metric_name)
   values ('analytics_test1'), ('analytics_test2');


   insert into business_hours_config
      (name, description, shift_by, shift_start_time, shift_end_time)
   values
      ('austin', 'Austin data center', '-06:00', '8:00', '16:00'),
      ('bangalore', 'Bangalore data center', '-09:30', '7:00', '17:00');

   insert into business_hours_group_assoc
      (business_hours_config_id, group_filter_id)
   values
      (
         (select id from business_hours_config where name='austin'),
         (select id from group_filter where name='austin')
      ),
      (
         (select id from business_hours_config where name='bangalore'),
         (select id from group_filter where name='bangalore')
      );

   insert into business_hours_metric_assoc
      (business_hours_config_id, metric_key)
   values
      (
         (select id from business_hours_config where name='austin'),
         (select metric_key from meta_metric where metric_name='analytics_test1')
      ),
      (
         (select id from business_hours_config where name='austin'),
         (select metric_key from meta_metric where metric_name='analytics_test2')
      ),
      (
         (select id from business_hours_config where name='bangalore'),
   ```

```
      (select metric_key from meta_metric where metric_name='analytics_test1')
   ),
   (
      (select id from business_hours_config where name='bangalore'),
      (select metric_key from meta_metric where metric_name='analytics_test2')
   );
```

4.5.7. Creating Nth Percentile Calculations

Zenoss Analytics allows you to create Nth percentile calculations which can be added to a report. The Nth percentile is the smallest value in the set of raw data with the property that n% of the data values are less than or equal to it. You need to define several parameters in order for the calculation to provide the targeted information you need:

- Value of percentile desired (e.g., 95th percentile)
- Number of days of data to aggregate (e.g., prior 30 days, this value is bounded by the `raw_retention_days` value)
- Business hours to calculate within (if defined, otherwise the default 24-hour "shift" is used)
- Job filter (also known as a group filter) to limit the calculation to certain devices (otherwise all devices are included)
- Metrics filter to limit the calculation to certain metrics (otherwise all metrics are included)

The creation of rows for the Nth percentile calculation is performed manually by inserting records into the reporting database. The tables of interest are described below:

- `nth_percentile_config`: main table defining the percentile jobs
  - `percentile`: nth value to calculate, e.g., the 95th
  - `period`: number of days worth of data to use for the nth percentile calculation
  - `name`: not used for nth percentile. Only exists for integration with an interface.
  - `description`: not used for nth percentile. Only exists for integration with an interface.
- `business_hours_config`: specifies time ranges that has been defined as a "business day" or "shift". See Section 4.5.6, “Creating Business Hours Aggregation” for more information about this table.
- `nth_percentile_business_hours_assoc`: relates `nth_percentile_config` records with `business_hours_config` records
  - `nth_percentile_config_id`: `nth_percentile_config` foreign key
  - `business_hours_config_id`: `business_hours` foreign key
- `group_filter`: optional filter used to reduce the set of devices included in the aggregation. See Section 4.5.6, “Creating Business Hours Aggregation” for more information about this table.
- `nth_percentile_group_assoc`: relates `nth_percentile_config` records with `group_filter` records
  - `nth_percentile_config_id`: `nth_percentile_config` foreign key
  - `group_filter_id`: `group_filter` foreign key
- `meta_metric`: optional filter used to reduce the set of metrics included in the aggregation. See Section 4.5.6, “Creating Business Hours Aggregation” for more information about this table.

- `nth_percentile_metric_assoc`: relates `nth_percentile_config` records with `meta_metric` records
  - `nth_percentile_config_id`: `nth_percentile_config` foreign key
  - `metric_key`: `meta_metric` foreign key

4.5.7.1. Inserting Records for Nth Percentile Calculations

Perform the following steps to insert records into the reporting database. The code in this procedure is for demonstration purposes only. This example assumes that you have two data centers (Austin and Bangalore) already defined and the metrics you want to filter on are called `analytics_test1` and `analytics_test2`:

1. Log in to the Analytics reporting database as the `root` user or a user with administrative permissions.

   ```
   zends -u root reporting
   ```

2. Enter the following command at the zends prompt substituting values that apply to your company's situation:

   ```
   zends> insert into nth_percentile_config
      (name, description, percentile, period)
   values
      ('percentile_test_1', 'first percentile test', 90, 10),
      ('percentile_test_2', 'second percentile test', 80, 20);

   insert into nth_percentile_business_hours_assoc
      (nth_percentile_config_id, business_hours_config_id)
   values
      (
         (select id from nth_percentile_config where percentile = 90 and period = 10),
         (select id from business_hours_config where name='austin')
      ),
      (
         (select id from nth_percentile_config where percentile = 80 and period = 20),
         (select id from business_hours_config where name='bangalore')
      );

   insert into nth_percentile_group_assoc
      (nth_percentile_config_id, group_filter_id)
   values
      (
         (select id from nth_percentile_config where percentile = 90 and period = 10),
         (select id from group_filter where name='austin')
      ),
      (
         (select id from nth_percentile_config where percentile = 80 and period = 20),
         (select id from group_filter where name='bangalore')
      );

   insert into nth_percentile_metric_assoc
      (nth_percentile_config_id, metric_key)
   values
      (
         (select id from nth_percentile_config where percentile = 90 and period = 10),
         (select metric_key from meta_metric where metric_name='analytics_test1')
      ),
      (
         (select id from nth_percentile_config where percentile = 90 and period = 10),
         (select metric_key from meta_metric where metric_name='analytics_test2')
      ),
      (
         (select id from nth_percentile_config where percentile = 80 and period = 20),
         (select metric_key from meta_metric where metric_name='analytics_test1')
      ),
   ```

```
   (
      (select id from nth_percentile_config where percentile = 80 and period = 20),
      (select metric_key from meta_metric where metric_name='analytics_test2')
   );
```

4.5.8. Projecting Other Aggregates

You can project Nth percentile values into the future by defining the number of days of past data to analyze along with the number of days into the future you'd like to predict. For example, you would like to predict the next 60 days of an Nth percentile calculation based on the last 30 days of data. This prediction will return a value for every day for the next 60 days. In order to accomplish this, a table (`projection_config`) is defined in the reporting database for the projection configuration of the future values.

The `projection_config` table has the following form:

- `name`: name to identify the projection (e.g., `polynomial_bangalore`, `harmonic_austin`)
- `description`: up to 255 characters to describe the projection
- `algorithm`: algorithm to use when performing calculation. Possible values are: `HARMONIC`, `POLYNOMIAL`.

  Each type of algorithm has its own associated optional parameters:

  - Harmonic (used when data follows an oscillating pattern)
    - Maximum iterations
    - Amplitude (initial guess value)
    - Angular frequency (initial guess value)
    - Phase (initial guess value)
  - Polynomial (used when data follows a curved pattern toward a maximum or minimum value, e.g., percentage of disk storage used over time)
    - Polynomial degree, maximum of 3
    - Coefficient 1, initial guess value, degree = 1
    - Coefficient 2, initial guess value, degree = 1
    - Coefficient 3, initial guess value, degree = 2
    - Coefficient 4, initial guess value, degree = 3
- `days_past`: integer indicating the number of past days to use in making the calculation
- `days_future`: integer indicating the number of days into the future to make projections
- `agg_column_name`: aggregation column name

Projections are run each day at the completion of aggregation and Nth percentile processing. Resultant tables from projection computations have the same structure as aggregation tables with the addition of a column with the timestamp of when the projection was performed. The naming convention of the projection table is in the form: `daily_<projection_name>_<metric_name>`

Each projection row may have one or several associated job filters and metric filters. A job filter reduces the device sample set by allowing you to create a device group and assigning devices to that group. Several job filters may be defined for a given projection. Each filter is additive and increases the number of the devices included in the sample set. If no job filters are associated with this filter, all devices are included in the sample set.

In addition to the group filter(s), you may also define and associate metric filters which restrict the metrics processed in the projection. If no metric filters are defined, all metrics are processed. Each metric processed is run in a separate thread. The projection calculations use data from the `raw_v2` tables. To accommodate the maximum flexibility in which data can be used, the metric name in this context is the name of the daily table without the preceding "`daily_`". For normal aggregated metrics, this equates to the metric name (e.g., `cpu__pct`). For specially aggregated (computed) data, such as business hours or Nth percentile, the metric name includes the computed name (e.g., `harmonic_austin_cpu__pct`)

4.5.8.1. Inserting Records for Nth Percentile Projections

Perform the following steps to insert records into the reporting database. The code in this procedure is for demonstration purposes only. This example assumes that you have two data centers (Austin and Bangalore) already defined and the metrics you want to filter on are called `analytics_test1` and `analytics_test2`:

1. Log in to the Analytics reporting database as the `root` user or a user with administrative permissions.

   ```
   zends -u root reporting
   ```

2. Enter the following command at the zends prompt substituting values that apply to your company's situation:

   ```
   zends> insert into projection_config
      (name, algorithm, days_past, days_future, agg_column_name, degree,
          curve_parameter_01,curve_parameter_02,curve_parameter_03)
   values
      ('polynomial_austin', 'POLYNOMIAL', 60, 30, 'fct_avg', 2, 1.0, 1.0, 1.0),
      ('polynomial_bangalore', 'POLYNOMIAL', 60, 30, 'fct_avg', 2, 1.0, 1.0, 1.0),
      ('harmonic_austin', 'HARMONIC', 60, 30, 'fct_avg', 0, 1.0, 1.0, 1.0),
      ('harmonic_bangalore', 'HARMONIC', 60, 30, 'fct_avg', 0, 1.0, 1.0, 1.0);

   insert into projection_business_hours_assoc
      (projection_config_id, business_hours_config_id)
   values
      (
         (select id from projection_config where name='polynomial_austin'),
         (select id from business_hours_config where name='austin')
      ),
      (
         (select id from projection_config where name='polynomial_bangalore'),
         (select id from business_hours_config where name='bangalore')
      ),
      (
         (select id from projection_config where name='harmonic_austin'),
         (select id from business_hours_config where name='austin')
      ),
      (
         (select id from projection_config where name='harmonic_bangalore'),
         (select id from business_hours_config where name='bangalore')
      );

   insert into projection_group_assoc
      (projection_config_id, group_filter_id)
   values
      (
         (select id from projection_config where name='polynomial_austin'),
         (select id from group_filter where name='austin')
      ),
      (
         (select id from projection_config where name='polynomial_bangalore'),
         (select id from group_filter where name='bangalore')
   ```

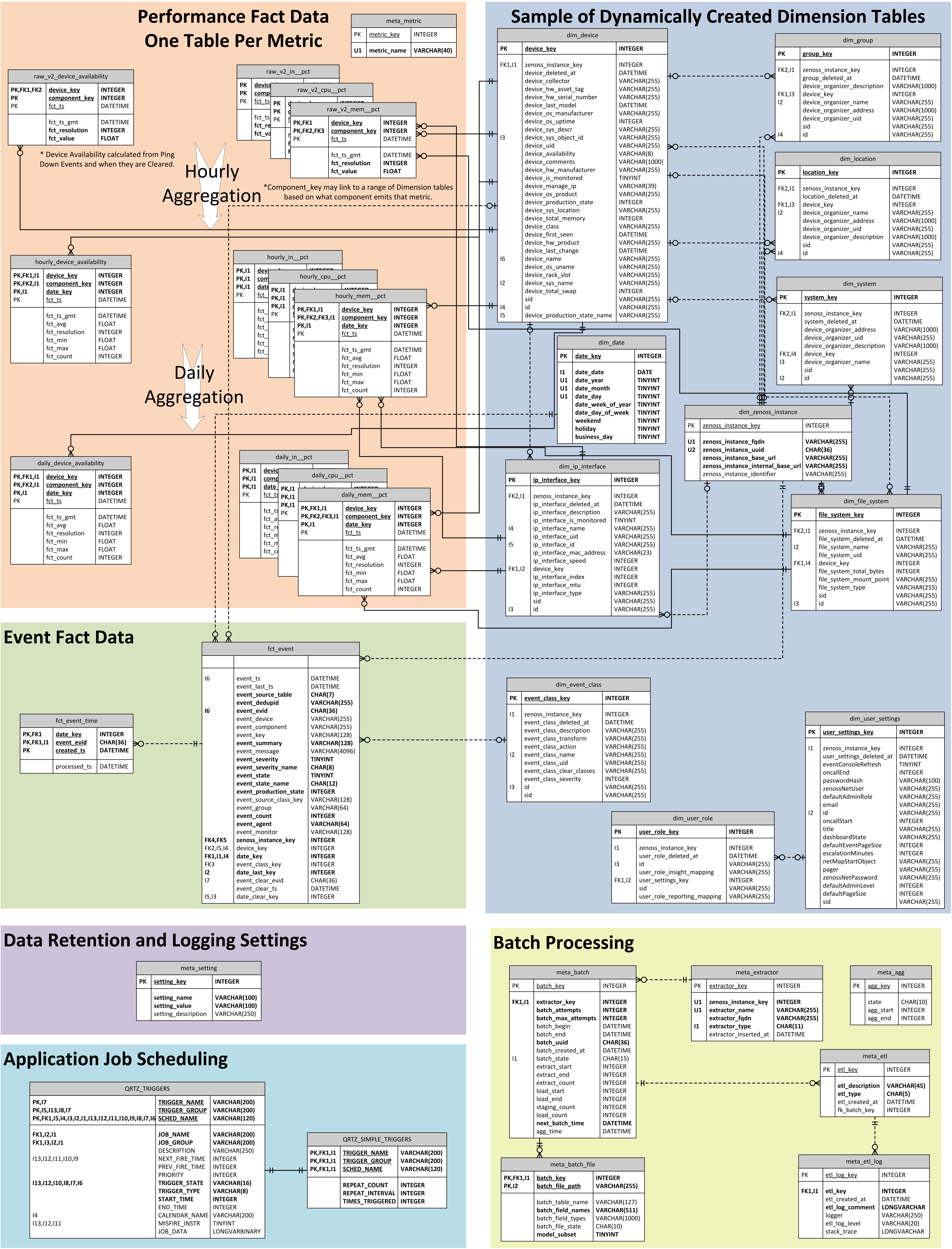
```
   ),
   (
      (select id from projection_config where name='harmonic_austin'),
      (select id from group_filter where name='austin')
   ),
   (
      (select id from projection_config where name='harmonic_bangalore'),
      (select id from group_filter where name='bangalore')
   );

insert into projection_metric_assoc
   (projection_config_id, metric_key)
values
   (
      (select id from projection_config where name='polynomial_austin'),
      (select metric_key from meta_metric where metric_name='analytics_test1')
   ),
   (
      (select id from projection_config where name='polynomial_austin'),
      (select metric_key from meta_metric where metric_name='analytics_test2')
   ),
   (
      (select id from projection_config where name='polynomial_bangalore'),
      (select metric_key from meta_metric where metric_name='analytics_test1')
   ),
   (
      (select id from projection_config where name='polynomial_bangalore'),
      (select metric_key from meta_metric where metric_name='analytics_test2')
   ),
   (
      (select id from projection_config where name='harmonic_austin'),
      (select metric_key from meta_metric where metric_name='analytics_test1')
   ),
   (
      (select id from projection_config where name='harmonic_austin'),
      (select metric_key from meta_metric where metric_name='analytics_test2')
   ),
   (
      (select id from projection_config where name='harmonic_bangalore'),
      (select metric_key from meta_metric where metric_name='analytics_test1')
   ),
   (
      (select id from projection_config where name='harmonic_bangalore'),
      (select metric_key from meta_metric where metric_name='analytics_test2')
   );
```

Appendix A. ERD Diagram for Analytics 4.3.1

A.1. Introduction

The following page shows the Entity-Relationship Diagram (ERD) for Zenoss Analytics 4.3.1. You will find the relationships shown between tables in the database as well as a sample of dynamically created dimension tables.



Performance Fact Data One Table Per Metric

meta_metric

PK	metric_key	INTEGER
U1	**metric_name**	**VARCHAR(40)**

raw_v2_device_availability

PK,FK1,FK2 **PK** PK	**device_key** **component_key** fct_ts	**INTEGER** **INTEGER** DATETIME
	fct_ts_gmt **fct_resolution** **fct_value**	DATETIME **INTEGER** **FLOAT**

* Device Availability calculated from Ping Down Events and when they are Cleared.

raw_v2_in__pct

raw_v2_cpu__pct

raw_v2_mem__pct

PK,FK1 **PK,FK2,FK3** PK	**device_key** **component_key** fct_ts	**INTEGER** **INTEGER** DATETIME
	fct_ts_gmt **fct_resolution** **fct_value**	DATETIME **INTEGER** **FLOAT**

Hourly Aggregation

*Component_key may link to a range of Dimension tables based on what component emits that metric.

hourly_device_availability

PK,FK1,I1 **PK,FK2,I1** **PK,I1** PK	**device_key** **component_key** **date_key** fct_ts	**INTEGER** **INTEGER** **INTEGER** DATETIME
	fct_ts_gmt fct_avg fct_resolution fct_min fct_max fct_count	DATETIME FLOAT INTEGER FLOAT FLOAT INTEGER

hourly_in__pct

hourly_cpu__pct

hourly_mem__pct

PK,FK1,I1 **PK,FK2,FK3,I1** **PK,I1** PK	**device_key** **component_key** **date_key** fct_ts	**INTEGER** **INTEGER** **INTEGER** DATETIME
	fct_ts_gmt fct_avg fct_resolution fct_min fct_max fct_count	DATETIME FLOAT INTEGER FLOAT FLOAT INTEGER

Daily Aggregation

daily_device_availability

PK,FK1,I1 **PK,FK2,I1** **PK,I1** PK	**device_key** **component_key** **date_key** fct_ts	**INTEGER** **INTEGER** **INTEGER** DATETIME
	fct_ts_gmt fct_avg fct_resolution fct_min fct_max fct_count	DATETIME FLOAT INTEGER FLOAT FLOAT INTEGER

daily_in__pct

daily_cpu__pct

daily_mem__pct

PK,FK1,I1 **PK,FK2,FK3,I1** **PK,I1** PK	**device_key** **component_key** **date_key** fct_ts	**INTEGER** **INTEGER** **INTEGER** DATETIME
	fct_ts_gmt fct_avg fct_resolution fct_min fct_max fct_count	DATETIME FLOAT INTEGER FLOAT FLOAT INTEGER

Sample of Dynamically Created Dimension Tables

dim_device

PK	**device_key**	**INTEGER**
FK1,I1	zenoss_instance_key	INTEGER
	device_deleted_at	DATETIME
	device_collector	VARCHAR(255)
	device_hw_asset_tag	VARCHAR(255)
	device_hw_serial_number	VARCHAR(255)
	device_last_model	DATETIME
	device_os_manufacturer	VARCHAR(255)
	device_os_uptime	INTEGER
	device_sys_descr	VARCHAR(255)
I3	device_sys_object_id	VARCHAR(255)
	device_uid	VARCHAR(255)
	device_availability	VARCHAR(8)
	device_comments	VARCHAR(1000)
	device_hw_manufacturer	VARCHAR(255)
	device_is_monitored	TINYINT
	device_manage_ip	VARCHAR(39)
	device_os_product	VARCHAR(255)
	device_production_state	INTEGER
	device_sys_location	VARCHAR(255)
	device_total_memory	INTEGER
	device_class	VARCHAR(255)
	device_first_seen	DATETIME
	device_hw_product	VARCHAR(255)
	device_last_change	DATETIME
I6	device_name	VARCHAR(255)
	device_os_uname	VARCHAR(255)
	device_rack_slot	VARCHAR(255)
I2	device_sys_name	VARCHAR(255)
	device_total_swap	INTEGER
	sid	VARCHAR(255)
I4	id	VARCHAR(255)
I5	device_production_state_name	VARCHAR(255)

dim_group

PK	**group_key**	**INTEGER**
FK2,I1	zenoss_instance_key	INTEGER
	group_deleted_at	DATETIME
	device_organizer_description	VARCHAR(1000)
FK1,I3	device_key	INTEGER
I2	device_organizer_name	VARCHAR(255)
	device_organizer_address	VARCHAR(1000)
	device_organizer_uid	VARCHAR(255)
	sid	VARCHAR(255)
I4	id	VARCHAR(255)

dim_location

PK	**location_key**	**INTEGER**
FK2,I1	zenoss_instance_key	INTEGER
	location_deleted_at	DATETIME
FK1,I3	device_key	INTEGER
I2	device_organizer_name	VARCHAR(255)
	device_organizer_address	VARCHAR(1000)
	device_organizer_uid	VARCHAR(255)
	device_organizer_description	VARCHAR(1000)
	sid	VARCHAR(255)
I4	id	VARCHAR(255)

dim_system

PK	**system_key**	**INTEGER**
FK2,I1	zenoss_instance_key	INTEGER
	system_deleted_at	DATETIME
	device_organizer_address	VARCHAR(1000)
	device_organizer_uid	VARCHAR(255)
	device_organizer_description	VARCHAR(1000)
FK1,I4	device_key	INTEGER
I3	device_organizer_name	VARCHAR(255)
	sid	VARCHAR(255)
I2	id	VARCHAR(255)

dim_date

PK	**date_key**	**INTEGER**
I1	**date_date**	**DATE**
U1	**date_year**	**TINYINT**
U1	**date_month**	**TINYINT**
U1	**date_day**	**TINYINT**
	date_week_of_year	**TINYINT**
	date_day_of_week	**TINYINT**
	weekend	**TINYINT**
	holiday	**TINYINT**
	business_day	**TINYINT**

dim_zenoss_instance

PK	zenoss_instance_key	INTEGER
U1	**zenoss_instance_fqdn**	**VARCHAR(255)**
U2	**zenoss_instance_uuid**	**CHAR(36)**
	zenoss_instance_base_url	**VARCHAR(255)**
	zenoss_instance_internal_base_url	**VARCHAR(255)**
	zenoss_instance_identifier	VARCHAR(255)

dim_ip_interface

PK	**ip_interface_key**	**INTEGER**
FK2,I1	zenoss_instance_key	INTEGER
	ip_interface_deleted_at	DATETIME
	ip_interface_description	VARCHAR(255)
	ip_interface_is_monitored	TINYINT
I4	ip_interface_name	VARCHAR(255)
	ip_interface_uid	VARCHAR(255)
I5	ip_interface_id	VARCHAR(255)
	ip_interface_mac_address	VARCHAR(23)
	ip_interface_speed	INTEGER
FK1,I2	device_key	INTEGER
	ip_interface_index	INTEGER
	ip_interface_mtu	INTEGER
	ip_interface_type	VARCHAR(255)
	sid	VARCHAR(255)
I3	id	VARCHAR(255)

dim_file_system

PK	**file_system_key**	**INTEGER**
FK2,I1	zenoss_instance_key	INTEGER
	file_system_deleted_at	DATETIME
I2	file_system_name	VARCHAR(255)
	file_system_uid	VARCHAR(255)
FK1,I4	device_key	INTEGER
	file_system_total_bytes	INTEGER
	file_system_mount_point	VARCHAR(255)
	file_system_type	VARCHAR(255)
	sid	VARCHAR(255)
I3	id	VARCHAR(255)

dim_event_class

PK	**event_class_key**	**INTEGER**
I1	zenoss_instance_key	INTEGER
	event_class_deleted_at	DATETIME
	event_class_description	VARCHAR(255)
	event_class_transform	VARCHAR(255)
	event_class_action	VARCHAR(255)
I2	event_class_name	VARCHAR(255)
	event_class_uid	VARCHAR(255)
	event_class_clear_classes	VARCHAR(255)
	event_class_severity	INTEGER
I3	id	VARCHAR(255)
	sid	VARCHAR(255)

dim_user_settings

PK	**user_settings_key**	**INTEGER**
I1	zenoss_instance_key	INTEGER
	user_settings_deleted_at	DATETIME
	eventConsoleRefresh	TINYINT
	oncallEnd	INTEGER
	passwordHash	VARCHAR(100)
	zenossNetUser	VARCHAR(255)
	defaultAdminRole	VARCHAR(255)
	email	VARCHAR(255)
I2	id	VARCHAR(255)
	oncallStart	INTEGER
	title	VARCHAR(255)
	dashboardState	VARCHAR(255)
	defaultEventPageSize	INTEGER
	escalationMinutes	INTEGER
	netMapStartObject	VARCHAR(255)
	pager	VARCHAR(255)
	zenossNetPassword	VARCHAR(255)
	defaultAdminLevel	INTEGER
	defaultPageSize	INTEGER
	sid	VARCHAR(255)

dim_user_role

PK	**user_role_key**	**INTEGER**
I1	zenoss_instance_key	INTEGER
	user_role_deleted_at	DATETIME
I3	id	VARCHAR(255)
	user_role_insight_mapping	VARCHAR(255)
FK1,I2	user_settings_key	INTEGER
	sid	VARCHAR(255)
	user_role_reporting_mapping	VARCHAR(255)

Event Fact Data

fct_event_time

PK,FK1 **PK,FK1,I1** **PK**	**date_key** **event_evid** **created_ts**	**INTEGER** **CHAR(36)** **DATETIME**
	processed_ts	DATETIME

fct_event

I6	event_ts	DATETIME
	event_last_ts	DATETIME
	event_source_table	**CHAR(7)**
	event_dedupid	**VARCHAR(255)**
I6	**event_evid**	**CHAR(36)**
	event_device	VARCHAR(255)
	event_component	VARCHAR(255)
	event_key	VARCHAR(128)
	event_summary	**VARCHAR(128)**
	event_message	VARCHAR(4096)
	event_severity	**TINYINT**
	event_severity_name	**CHAR(8)**
	event_state	**TINYINT**
	event_state_name	**CHAR(12)**
	event_production_state	**INTEGER**
	event_source_class_key	VARCHAR(128)
	event_group	VARCHAR(64)
	event_count	**INTEGER**
	event_agent	**VARCHAR(64)**
	event_monitor	VARCHAR(128)
FK4,FK5	**zenoss_instance_key**	**INTEGER**
FK2,I5,I4	device_key	INTEGER
FK1,I1,I4	**date_key**	**INTEGER**
FK3	event_class_key	INTEGER
I2	**date_last_key**	**INTEGER**
I7	event_clear_evid	CHAR(36)
	event_clear_ts	DATETIME
I5,I3	date_clear_key	INTEGER

Data Retention and Logging Settings

meta_setting

PK	**setting_key**	**INTEGER**

	setting_name
	setting_value
	setting_description
		VARCHAR(100)
		VARCHAR(100)
		VARCHAR(250)

Application Job Scheduling

QRTZ_TRIGGERS

PK,I7	**TRIGGER_NAME**	**VARCHAR(200)**
PK,I5,I13,I8,I7	**TRIGGER_GROUP**	**VARCHAR(200)**
PK,FK1,I5,I4,I3,I2,I1,I13,I12,I11,I10,I9,I8,I7,I6	**SCHED_NAME**	**VARCHAR(120)**
FK1,I2,I1	**JOB_NAME**	**VARCHAR(200)**
FK1,I3,I2,I1	**JOB_GROUP**	**VARCHAR(200)**
	DESCRIPTION	VARCHAR(250)
I13,I12,I11,I10,I9	NEXT_FIRE_TIME	INTEGER
	PREV_FIRE_TIME	INTEGER
	PRIORITY	INTEGER
I13,I12,I10,I8,I7,I6	**TRIGGER_STATE**	**VARCHAR(16)**
	TRIGGER_TYPE	**VARCHAR(8)**
	START_TIME	**INTEGER**
	END_TIME	INTEGER
I4	CALENDAR_NAME	VARCHAR(200)
I13,I12,I11	MISFIRE_INSTR	TINYINT
	JOB_DATA	LONGVARBINARY

QRTZ_SIMPLE_TRIGGERS

PK,FK1,I1	**TRIGGER_NAME**	**VARCHAR(200)**
PK,FK1,I1	**TRIGGER_GROUP**	**VARCHAR(200)**
PK,FK1,I1	**SCHED_NAME**	**VARCHAR(120)**
	REPEAT_COUNT	**INTEGER**
	REPEAT_INTERVAL	**INTEGER**
	TIMES_TRIGGERED	**INTEGER**

Batch Processing

meta_batch

PK	batch_key	INTEGER
FK1,I1	**extractor_key**	**INTEGER**
	batch_attempts	**INTEGER**
	batch_max_attempts	**INTEGER**
	batch_begin	DATETIME
	batch_end	DATETIME
	batch_uuid	**CHAR(36)**
	batch_created_at	DATETIME
I1	batch_state	CHAR(15)
	extract_start	INTEGER
	extract_end	INTEGER
	extract_count	INTEGER
	load_start	INTEGER
	load_end	INTEGER
	staging_count	INTEGER
	load_count	INTEGER
	next_batch_time	**DATETIME**
	agg_time	DATETIME

meta_extractor

PK	extractor_key	INTEGER
U1	**zenoss_instance_key**	**INTEGER**
U1	**extractor_name**	**VARCHAR(255)**
	extractor_fqdn	**VARCHAR(255)**
I1	**extractor_type**	**CHAR(11)**
	extractor_inserted_at	DATETIME

meta_agg

PK	agg_key	INTEGER
	state	CHAR(10)
	agg_start	INTEGER
	agg_end	INTEGER

meta_etl

PK	etl_key	INTEGER
	etl_description	**VARCHAR(45)**
	etl_type	**CHAR(5)**
	etl_created_at	DATETIME
	fk_batch_key	INTEGER

meta_batch_file

PK,FK1,I1	**batch_key**	**INTEGER**
PK,I2	**batch_file_path**	**VARCHAR(255)**
	batch_table_name	VARCHAR(127)
	batch_field_names	**VARCHAR(511)**
	batch_field_types	VARCHAR(1000)
	batch_file_state	CHAR(10)
	model_subset	**TINYINT**

meta_etl_log

PK	etl_log_key	INTEGER
FK1,I1	**etl_key**	**INTEGER**
	etl_created_at	DATETIME
	etl_log_comment	**LONGVARCHAR**
	logger	VARCHAR(250)
	etl_log_level	VARCHAR(20)
	stack_trace	LONGVARCHAR