Managing Prime Analytics

Prime Analytics administrators perform many tasks that maintain the health and performance of the Prime Analytics BI platform and databases. Using the BI platform Administration Console, administrators can manage users and user roles, database connections, and database query schedules. Additional Administration functions include backing up the Prime Analytics BI platform and database and, if needed, restoring the BI platform and database.

Administrative functions and tasks are covered in the following topics:

Logging Into the BI Platform Administration Console

Accessing the Prime Analytics Administration console requires Admin security access. To log into the BI Administration Console:

Step 1 Launch your web browser, then enter one of the following in the browser URL field:

- ***<Prime Analytics BI Server>*:8080/pentaho/Admin**—Launches the Administration Console directly.
- ***<Prime Analytics BI Server>*:8080**—Displays the User Console. On the User Console under Administration, click **Settings**.

Step 2 At the login window, enter the Admin username and password, then click **OK**.

The Prime Analytics Administration Console appears. The console is divided into two areas:

- Admin Services—Includes the following administrative services:
  - Clean Repository—Removes files from the content repository that are more than 180 days old.
  - Scheduler Admin—Allows to manage the scheduler. See "Managing BI Server Schedules" procedure on page 8-10.
  - Manage Users and Roles—Allows you to manages Prime Analytics BI platform users and roles. (see Managing Users, page 8-2), database connections, (see Managing the Database Connections, page 8-7), services (see Managing BI Server Services, page 8-9, and schedules (see Managing BI Server Schedules, page 8-10).
  - Permissions—Allows you to set execute and overwrite permissions for content for users and roles.
  - Subscription Admin—Allows you to manage subscription content and schedules.
- Refresh—Includes the following items:
  - Plugin Adapter—Refreshes all the BI platform plugin applications.
  - Subscription publisher—Publishes imported schedules and content.
  - Update solution repository—Reads all of the solution files and regenerates the repository index.
  - Reporting Metadata—Refreshes the metadata used for ad hoc reporting.
  - Global Actions—Executes all global system actions defined in pentaho.xml.

Managing Users

BI platform user management tasks are performed using the Prime Analytics Administration Console > Manage Users and Roles > Users & Roles tab.

By default, the following users roles are provided with the Prime Analytics installation:

- Admin—Has full create, read, update, and delete access to all user and administrator functions including user and user role management.
- Designer—Has full create, read, update, and delete access to all User Console features including access the data source functionality. Cannot access the Administrator Console.
- Developer—Has full or partial create, read, update, and delete access for dashboards and roles assigned by the administrator. Cannot access the Administration Console and cannot access or edit data source functionality.
- Authenticated—A user who has logged in.
- Anonymous—A user who has not logged in.

In addition to default user roles, Table 8-1 lists the users provided with Prime Analytics by default.

Table 8-1 Default Users

Username	Authority
joe	Authenticated, Admin
admin	User required to manage users and user roles.

Table 8-1 Default Users (continued)

Username	Authority
katy	Authenticated, Designer
bill	Authenticated, Developer
sam	Authenticated

BI platform user management is covered in the following topics:

Changing Default User Passwords

By default, Prime Analytics ships with the following user passwords:

- joe/password
- admin/password
- katy/password
- bill/password
- sam/password

Changing the default passwords is something you should do following installation. Changing the password for users joe, katy, bill, and sam is performed by completing the "Editing BI Platform User Information" procedure on page 8-5.

The admin user is required to access the Manage Users and Roles window. To change the default admin password, complete the following steps:

Step 1 Login as root users.

Step 2 Stop the BI platform:

```
service biplatform stop
```

Step 3 Login as the bipuser.

Step 4 Change to the administration-console directory:

```
cd $PA_HOME/biplatform/administration-console
```

Step 5 Enter the new password:

```
java -cp lib/jetty-6.1.2.jar:lib/jetty-util-6.1.9.jar org.mortbay.jetty.security.Password
<user> <newpassword>
```

for example,

```
$ java -cp lib/jetty-6.1.2.jar:lib/jetty-util-6.1.9.jar
org.mortbay.jetty.security.Password admin newpassword
newpassword
newpassword
OBF:1uo91vn61ymf1yt41v1p1ym71v2p1yti1ylz1vnw1unp
MD5:5e9d11a14ad1c8dd77e98ef9b53fd1ba
CRYPT:adx.wc8vu/gJU
```

Step 6 Change to the config directory:

```
cd resource/config
```

Step 7 In login.properties, enter the new OBF entry:

admin: OBF:1uo91vn61ymf1yt41v1p1ym71v2p1yti1ylz1vnw1unp,admin

Step 8 Log in as the root user and start the BI platform:

```
service biplatform start
```

Creating New BI Platform Users

To create new BI platform users:

Step 1 Log into the Prime Analytics Administration Console. See Logging Into the BI Platform Administration Console, page 8-1.

Step 2 Under Admin Services, click **Manage Users and Roles**.

Step 3 At the login screen, enter the user, admin, and the admin user password.

Step 4 On the Users & Roles tab, click the **Users** icon.

Step 5 Next to the Users list title, click **+** (Add User).

Step 6 In the Add User dialog box, enter the following:

- User Name
- Password
- Password Confirmation
- Description

Step 7 Click **OK**.

The new user is added to the Users list.

Step 8 Select the user you just created.

Step 9 Next to Assigned Roles, click **+** (Assign Role).

Step 10 In the Assign Role dialog box, choose the roles you want to assign to the user from the Available Roles list.

Step 11 Click **OK**.

Editing BI Platform User Information

To edit BI platform user information:

Step 1 Log into the Prime Analytics Administration Console. See Logging Into the BI Platform Administration Console, page 8-1.

Step 2 Under Admin Services, click **Manage Users and Roles**.

Step 3 At the login screen, enter the user, admin, and the admin user password.

Step 4 On the Users & Roles tab, click the **Users** icon.

Step 5 Under Users, choose the user you want to edit.

Step 6 Under Details, edit any of the following information:

- User Name
- Password
- Password Confirmation
- Description
- Assigned Roles—Click **Assign Roles** or **Unassign Roles** to add or remove roles to and from the user.

Step 7 Click **OK**.

Tip If you have many users, you can find a specific user by entering the first few letters of the user's name in the Filter box. A list of names matching your entry appears.

Deleting BI Platform Users

To delete BI platform users:

Step 1 Log into the Prime Analytics Administration Console. See Logging Into the BI Platform Administration Console, page 8-1.

Step 2 Under Admin Services, click **Manage Users and Roles**.

Step 3 At the login screen, enter the user, admin, and the admin user password.

Step 4 On the Users & Roles tab, click the **Users** icon.

Step 5 Under Users, choose the user you want to delete.

Step 6 Click **X** (Delete Users).

Step 7 On the confirmation, click **OK**.

Creating New BI Platform User Roles

To create new BI platform user role:

Step 1 Log into the Prime Analytics Administration Console. See Logging Into the BI Platform Administration Console, page 8-1.

Step 2 Under Admin Services, click **Manage Users and Roles**.

Step 3 At the login screen, enter the user, admin, and the admin user password.

Step 4 On the Users & Roles tab, click **Roles**.

Step 5 Next to the Roles title, click **+** (Add Role).

Step 6 In the Add Role dialog box, enter the following:

- Role Name
- Description

Step 7 Click **OK**.

The new role is added to the Roles list.

Editing BI Platform User Roles

To edit a BI platform user role:

Step 1 Log into the Prime Analytics Administration Console. See Logging Into the BI Platform Administration Console, page 8-1.

Step 2 Under Admin Services, click **Manage Users and Roles**.

Step 3 At the login screen, enter the user, admin, and the admin user password.

Step 4 On the Users & Roles tab, click **Roles**.

Step 5 Under Roles, choose the role you want to edit.

Step 6 Under Details, edit any of the following information:

- Role Name
- Description

Step 7 Click **OK**.

Tip If you have many roles, you can find a specific role by entering the first few letters of the role name in the Filter box. A list of roles matching your entry appears.

Deleting BI Platform User Roles

To delete a BI platform user role:

Step 1 Log into the Prime Analytics Administration Console. See Logging Into the BI Platform Administration Console, page 8-1.

Step 2 Under Admin Services, click **Manage Users and Roles**.

Step 3 At the login screen, enter the user, admin, and the admin user password.

Step 4 On the Users & Roles tab, click **Roles**.

Step 5 Under Roles, choose the role you want to delete.

Step 6 Next to Assigned Roles, click **X** (Delete Roles).

Step 7 On the confirmation, click **OK**.

Managing the Database Connections

The BI platform Administration Console allows you to manage all connections to Prime Analytics databases, including access drivers and URLs, usernames and password required for access, maximum active connections, number of idle connections, validation queries, and other properties. Database connection management tasks are covered in the following topics:

- Creating a New Database Connection, page 8-7
- Editing Database Connections, page 8-8
- Deleting Database Connections, page 8-9

Creating a New Database Connection

To create a new database connection:

Step 1 Log into the Prime Analytics Administration Console. See Logging Into the BI Platform Administration Console, page 8-1.

Step 2 Under Admin Services, click **Manage Users and Roles**.

Step 3 At the login screen, enter the user, admin, and the admin user password.

Step 4 Click the **Database Connections** tab.

Step 5 Next to Database Connections, click **+** (Add Database Connection).

Step 6 In the Add Database Connection dialog box, enter the following:

- Name—Enter a name for the database connection.
- Driver Class—Enter the database driver class. The database driver name you select depends on the type of database you are accessing. For example, org.hsqldb.jdbcDriver is a sample driver name for a HypersonicSQL database. The default driver is com.primeanalytics.Driver. Other drivers can be added using the "Installing the JDBC Drivers" procedure on page 4-1.
- User Name—Enter the username required to use the database

- Password—Enter the username password.
- URL—Enter the URL used to access the database. This is the URL of your database. For example, jdbc:hsqldb:hsql://localhost/sampledata. JDBC establishes a connection to a SQL-based database and sends and processes SQL statements.

Step 7 Click Test to test your connection. If the test is successful, continue with the next step. If not, verify the information in the previous step and test again.

Step 8 Click the **Advanced** icon.

Step 9 As needed, enter the following information

- Maximum Active Connections—The maximum number of active database connections that can be allocated from this pool at the same time.
- Number of Idle Connections—The maximum number of that can sit idle in this pool at the same time.
- Validation Query—The SQL query that can be used by the pool to validate connections before they are returned to the application. If specified, this query must be an SQL SELECT statement that returns at least one row.
- Wait—The maximum number of milliseconds that the pool will wait (when there are no available connections) for a connection to be returned before throwing an exception.

Note If any fields are left empty, the BI platform uses the default values stored in the pentaho.xml.

Step 10 Click **OK**.

The new database connection is added to the Database Connections list.

Editing Database Connections

To edit BI platform database connections:

Step 1 Log into the Prime Analytics Administration Console. See Logging Into the BI Platform Administration Console, page 8-1.

Step 2 Under Admin Services, click **Manage Users and Roles**.

Step 3 At the login screen, enter the user, admin, and the admin user password.

Step 4 Click the **Database Connections** tab.

Step 5 Under Database Connections, choose the database connection you want to edit.

Step 6 Edit any of the following database properties. For property descriptions, see Creating a New Database Connection, page 8-7.

- Name
- Driver Class
- User Name
- Password
- URL

Step 7 Click the **Advanced** icon.

Step 8 As needed, edit the following information

- Maximum Active Connections
- Number of Idle Connections
- Validation Query
- Wait

Step 9 Click **Test** to verify that your edits have not invalidated the connection. If the test is successful, continue with the next step. If not, verify the information you entered and test again.

Deleting Database Connections

To delete BI platform database connections:

Step 1 Log into the Prime Analytics Administration Console. See Logging Into the BI Platform Administration Console, page 8-1.

Step 2 Under Admin Services, click **Manage Users and Roles**.

Step 3 At the login screen, enter the user, admin, and the admin user password.

Step 4 Click the **Database Connections** tab.

Step 5 Under Database Connections, choose the database connection you want to delete.

Step 6 Next to Database Connections, click **X** (Delete Database Connection).

Step 7 On the confirmation, click **OK**.

Managing BI Server Services

The Administration Console Services tab allows you to manage schedules and refresh the BI Server settings.

To manage BI platform server settings:

Step 1 Log into the Prime Analytics Administration Console. See Logging Into the BI Platform Administration Console, page 8-1.

Step 2 Under Admin Services, click **Manage Users and Roles**.

Step 3 At the login screen, enter the user, admin, and the admin user password.

Step 4 Click the **Services** tab.

Perform any of the following services:

- Schedule—Schedules the daily removal of files created in the content repository located in /pentaho-solution/system/content that are over 180 days old. To change the number of days, edit the solution file clean_repository.xaction located in /pentaho-solution/admin. To change the recurrence, edit the solution file schedule-clean.xaction located in /pentaho-solution/admin.

- Execute—Removes files created in the content repository located in /pentaho-solution/system/content that are over 180 days old. To change, the number of days, edit the solution file clean_repository.xaction located in /pentaho-solution/admin.
- Refresh—Reads all solution files and regenerates the repository index.
- Reset Permissions—Resets all permissions.

Managing BI Server Schedules

The Administration Console Scheduler tab allows you to create, update, delete, run, suspend, and resume one or more private and public schedules, You can also suspend and resume the BI server scheduler itself. In the context of the BI platform, a schedule is a time (or group of times) associated with an action sequence (or group of action sequences). In many instances, the output of an action sequence associated with a public schedule is a report, for example, a device domain report to which a network administrator can subscribe. As the administrator, the schedule (or schedules) you designate determines when the scheduler allows the action sequence to run. Regular schedules are ad hoc, non-subscription schedules, which are associated with one action sequence only.

In addition to associating a time (or group of times in the case of a repeating schedule) with an action sequence or group of action sequences, the public schedule is also associated with a user's My Workspace. When an action sequence runs on its defined schedule, the output of the action sequence (typically a report) is archived in the My Workspace of the user(s) who have subscribed to that action sequence. This allows the subscribers to view the output of the action sequence (the report) at any time following its execution.

Management of server schedules is covered in the following topics:

- Creating BI Server Schedules, page 8-10
- Managing Schedules, page 8-11

Creating BI Server Schedules

You create BI server schedules associated with your action sequences using the Schedule Creator. The Schedule Creator allows to enter schedules without having to know cron syntax, although you can enter cron expressions in the Scheduler Creator if you prefer.

To create a BI server schedule:

Step 1 Log into the Prime Analytics Administration Console. See Logging Into the BI Platform Administration Console, page 8-1.

Step 2 Under Admin Services, click **Manage Users and Roles**.

Step 3 At the login screen, enter the user, admin, and the admin user password.

Step 4 Click the **Scheduler** tab.

Step 5 In the Schedule Creator dialog box Schedule tab, enter the following information:

- Public Schedule—Check if you want users to be able to access the schedule.
- Name—Enter a schedule name, for example, Monthly Device Availability Report.
- Group—Enter a group associated with the schedule, for example, Western Region.

- Description—Enter a description of the schedule. for example, Schedule runs on the first of each month, or Schedule runs on Monday of each week.
- Recurrence Type—Enter the frequency at which you want the action sequence to occur. You can:
  - Schedule the sequence to run once at a particular date and time,
  - Schedule the sequence to recur at regular intervals. The intervals can be expressed in seconds, minutes, hours, days, weeks, months, year.
  - Enter a cron string.

  The options change depending on the type of recurrence you select.

Step 6 Click **OK**.

Step 7 Click the **Selected Files** tab.

Step 8 Click the + (Select Item) icon.

Step 9 In the Select dialog box, navigate to the directory and select file(s) on which you want the action sequence to occur.

- To open a directory, choose the directory and click **Select**.
- To select a file(s), choose the file(s) and click **OK**.

Step 10 Click OK to close the Schedule Creator dialog box.

Managing Schedules

As you create new schedules, the schedules are added to the Scheduler tab list box. The schedule list provides the following information for each schedule:

- Name—The schedule name.
- Group—The group assigned to the schedule
- State—The schedule state, either Normal or Suspended.
- Next Fire Time—The next date and time the schedule will run.
- Last Fire Time—The last date and time the schedule ran.

Use the Schedule toolbar controls, listed in Table 8-2, to perform schedule management actions.

Table 8-2 Schedule Actions

Action	Description
Create	Creates a new schedule
Edit	Allows you to edit the schedule details.
Delete	Deletes a specified schedule if it is not currently executing. If the schedule is currently executing, it continues to execute but no new instances are run.
Suspend	Pauses a specified schedule. After the job is suspended, you must click Resume to start it again.
Resume	Resumes a previously suspended schedule. After the schedule is resumed the scheduler applies misfire rules, if needed.
Run	Runs a schedule immediately.

Table 8-2 Schedule Actions (continued)

Action	Description
Refresh	Refreshes the list of schedules.
Filter by	Allows you to search for a specific schedule by group name.

Enabling SSL on the BI Platform

The following procedures tell you how to enable Secure Socket Layer (SSL) to enhance security and enable encrypted communications between Prime Analytics client web browsers and the BI platform server, including the administration and user consoles. Complete the following procedures in order:

- Creating an SSL Certificate, page 8-12
- Installing the Certificate, page 8-13
- Configuring Tomcat, page 8-14
- Configuring the Jetty Server with the SSL Certificate, page 8-14
- Configuring the Prime Analytics Properties File, page 8-15
- Validating SSL, page 8-16

Creating an SSL Certificate

The following steps tell you how to create a self-signed SSL certificate using the keytool that comes with the Java Development Kit. (In a production environment, obtain a certificate from a trusted certification authority.)

To generate an SSL certificate:

Step 1 Log into the Prime Analytics BI platform server as the root user.

Step 2 Stop the BI platform server:

```
service biplatform stop
```

Step 3 Switch to the bipuser:

```
su - bipuser
```

Step 4 Navigate to bipuser home directory:

```
cd $HOME
```

Step 5 Generate the SSL key:

```
keytool -genkey -alias tomcat -keyalg RSA
```

Enter the following parameters:

- Enter keystore password—Enter **changeit.**
- Re-enter new password—Enter **changeit.**
- What is your first and last name?—Enter **localhost**. If you do not enter localhost, the HostnameVerifier will fail.

- Enter key password for <tomcat>—Press **Enter/Return** to use same password as the one entered for keystore.

For example:

```
keytool -genkey -alias tomcat -keyalg RSA
Enter keystore password: changeit
Re-enter new password: changeit
What is your first and last name?
  [Unknown]:  localhost
What is the name of your organizational unit?
  [Unknown]:
What is the name of your organization?
  [Unknown]:
What is the name of your City or Locality?
  [Unknown]:
What is the name of your State or Province?
  [Unknown]:
What is the two-letter country code for this unit?
  [Unknown]:
Is CN=localhost, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown correct?
  [no]:  yes
Enter key password for <tomcat>
        (RETURN if same as keystore password):changeit
```

Step 6 Continue with the "Installing the Certificate" procedure on page 8-13.

Installing the Certificate

Complete the following steps to install the certificate created in Creating an SSL Certificate, page 8-12.

Step 1 As the bipuser, enter the following command in /home/bipuser:

```
keytool -export -alias tomcat -file tomcat.cer -storepass changeit -keypass changeit
-keystore .keystore
```

Step 2 Switch to the root user:

```
su - root
```

Step 3 Change to the $JAVA_HOME/jre/lib/security directory and enter the following command:

```
keytool -import -alias tomcat -file /home/bipuser/tomcat.cer -keystore cacerts -storepass
changeit
```

Step 4 Confirm that the tomcat entry in /home/bipuser/.keystore is the same entry that is in $JAVA_HOME/jre/lib/security/cacerts. To compare the fingerprints:

a. Execute the following in /home/bipuser:

```
keytool -list -keystore .keystore
Output of keytool -list -keystore .keystore
Keystore type: jks
Keystore provider: SUN
Your keystore contains 1 entry
tomcat, Mar 1, 2007, keyEntry,
Certificate fingerprint (MD5):
xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx
```

b. Note the Tomcat fingerprint entry, then enter the following in $JAVA_HOME/jre/lib/security:

```
keytool -list -keystore cacerts
Output of keytool -list -keystore cacerts
Keystore type: jks
Keystore provider: SUN
Your keystore contains n entries
entries omitted
tomcat, Mar 1, 2007, trustedCertEntry,
Certificate fingerprint (MD5):
xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx
entries omitted
```

c. Verify that the Tomcat entry in the cacerts fingerprint is the same as the Tomcat entry in .keystore.

Step 5 Continue with the "Configuring Tomcat" procedure on page 8-14.

Configuring Tomcat

Complete the following steps to configure Tomcat:

Step 1 Switch to the bipuser:

```
su - bipuser
```

Step 2 Change to the following directory:

```
cd $PA_HOME/biplatform/biserver-ce/tomcat/conf/
```

Step 3 Open server.xml with a text editor and add following entry:

```
<!-- Define a SSL Coyote HTTP/1.1 Connector on port 8443 -->
<Connector port="8443" maxThreads="200"
           scheme="https" secure="true" SSLEnabled="true"
           keystoreFile="/home/bipuser/.keystore" keystorePass="changeit"
           clientAuth="false" sslProtocol="TLS"/>
```

If needed, modify port, keystoreFile and keystorePass parameters.

Step 4 Continue with the "Configuring the Jetty Server with the SSL Certificate" procedure on page 8-14.

Configuring the Jetty Server with the SSL Certificate

Complete the following steps to configure the Jetty server with the SSL certificate.

Step 1 As the bipuser, navigate to bipuser home directory:

```
cd $HOME
```

Step 2 Enter the following command:

```
Keytool -keystore .keystore -alias jetty -genkey -keyalg RSA
```

You will be prompted to enter certificate information and passwords to protect the keystore and its keys. The only required information is the fully qualified server host name at the first and last name prompt. For example:

```
Enter keystore password:  changeit
What is your first and last name?
  \[Unknown\]:  localhost
What is the name of your organizational unit?
  \[Unknown\]:
What is the name of your organization?
  \[Unknown\]:
What is the name of your City or Locality?
  \[Unknown\]:
What is the name of your State or Province?
  \[Unknown\]:
What is the two-letter country code for this unit?
  \[Unknown\]:
Is CN=localhost, OU=Unknown, O=Unknown,
L=Unknown, ST=Unknown, C=Unknown correct?
  \[no\]:  yes
Enter key password for <jetty>
        (RETURN if same as keystore password): changeit
```

Note If the keystore already exists in $PA_HOME/biplatform/administration-console/resource/config you can skip the last two steps.

Step 3 After the certificate is created, you might need to locate the file on the disk. This generally is a keystore file in your home directory.

Step 4 Move the keystore file into $PA_HOME/biplatform/administration-console/ resource/config.

This is the default keystore location. If you place the keystore in a different location, be sure to keep this in mind.

Step 5 Continue with the "Configuring the Prime Analytics Properties File" procedure on page 8-15.

Configuring the Prime Analytics Properties File

Complete the following steps to configure the Prime Analytics Properties file:

Step 1 As the bipuser, switch to the following directory:

```
cd $PA_Home/biplatform/administration-console/resource/config
```

Step 2 Open the console.properties file in a text editor and change **console.ssl.enabled** from false to **true**.

By default, the keystore and trustore path is resource/config and password for them is changeit. If you have something different you can edit the SSL section for the correct value. Remember that the default port for the administration console https is 8043, If you want to change it, you can enter the change in the properties file.

Step 3 Switch to the following directory:

```
$PA_Home/biplatform/biserver-ce/tomcat/webapps/pentaho/WEB-INF/classes
```

Step 4 Open the adminprotocol.properties file in a text editor and change the following properties:

- ADMINISTRATION_CONSOLE_PROTOCOL—Change to **https**.
- ADMINISTRATION_CONSOLE_PORT—Enter the same value that was entered in console.ssl.port.number in console.properties.

For example:

```
ADMINISTRATION_CONSOLE_PROTOCOL=https
ADMINISTRATION_CONSOLE_PORT=8143
```

The Prime Analytics Administration Console is now ready to use with SSL.

Step 5 Continue with the "Validating SSL" procedure on page 8-16.

Validating SSL

Complete the following steps to validate SSL with Prime Analytics:

Step 1 Switch to the root user.

```
su - root
```

Step 2 Start the BI platform server:

```
service biplatform start
```

Step 3 Enter following URLs in a browser window:

- User Console: https://<server>:8443
- Administration Console: https://<server>:8443/pentaho/Admin

Step 4 Click **Manage Users and Roles** and verify that the Administration console is displayed.

Customizing Server Parameters

The Prime Analytics database server parameters are stored in a file named postgresql.conf. This file is stored in the root of your database data directory:

```
$PGDATA/postgresql.conf
```

Because TruCQ is built on PostgreSQL, all PostgreSQL configuration parameters are valid. For information about PostgreSQL parameters, see *http://www.postgresql.org/docs/8.3/static/runtime-config.html*

Note The postgresql.conf file has several logging parameters that you might want to adjust, For information, see TruCQ Logging for Problem Resolution, page 8-23.

Enabling and Restricting Network Access

Prime Analytics CQ engine defaults configuration allow anyone on the local machine to access the database, but access from anywhere else is blocked. If you want to restrict access or open it to more systems over the network, see the instructions at:

http://www.postgresql.org/docs/8.3/static/client-authentication.html.

In the default installation, PGDATA is set to /var/opt/primeanalytics/data. Therefore, you must modify $PGDATA/pg_hba.conf and the listen_addresses setting in $PGDATA/postgresql.conf. You might also need to adjust any active system firewalls to allow traffic through them.

TruCQ Server Parameters

Some Prime Analytics server configuration parameters are specific to TruCQ. They control TruCQ behavior. You can modify TruCQ parameters using normal PostgreSQL configuration interfaces:

- Editing the postgresql.conf file
- Using SET command
- Using the pg_settings system view.

Table 8-3 describes the TruCQ general option settings.

Table 8-3 TruCG General Option Settings

Option	Type	Default	Description
enable_cq	boolean	true	Enables CQ functionality in TruCQ. If disabled, TruCQ behaves like standard PostgreSQL: continuous queries are not allowed and stream processing does not occur. Disabling enable_cq allows streams to be created and altered in ways that might not be possible with enable_cq enabled. This can be useful when restoring a database.
cq_block_enqueue	boolean	true	Controls how TruCQ handles internal communication queues. These queues buffer communications between various TruCQ processes. If stream tuples arrive at a very high rate, the buffers might fill up. In the default configuration, TruCQ blocks streams until buffer space is available. If the parameter is disabled, TruCQ will drop data.
cq_cursor_block_enqueue	boolean	false	Controls how TruCQ handles full cursor communication queues. These queues buffer the results returned by cursors on continuous queries. If clients do not promptly fetch results from a cursor, the buffer might fill up. By default, TruCQ drops the newest data tuples when this occurs. This parameter enables the blocking. Enable this parameter when are running performance analyses. If not enabled, your results might be inflated because not all the produced data will be properly consumed.
cq_log_block_enqueue	boolean	false	Enables logging on blocked insertions into internal queues. The log message format is: `enqueue tuple is blocking on tuplechannel = {16349, 0, 1}`

Table 8-3 TruCG General Option Settings (continued)

Option	Type	Default	Description
cq_copy_ignore_errors	boolean	false	Specifies how COPY should handle stream input errors. By default, COPY in PostgreSQL aborts the current transaction when input data errors occur. If this parameter is enabled, TruCQ makes an exception for COPY commands loading data into streams. Erroneous input lines are dropped and logged. This allows the COPY to proceed. Correcting the data to avoid format errors is recommended. However, you can use this parameter as a workaround for erroneous data. If you use this workaround, actively monitor the logs to keep track of how much erroneous data you are receiving, then change your policies as needed.
cq_database	string	cqdb	Specifies the name of the database in which continuous queries are evaluated. You can use only one database for CQ data.
cq_telescope_view_enable	boolean	false	Enables creation of telescoping views.
cq_replication_enqueue_enable	boolean	true	Enables queuing of replicator work in a replicator catalog tables while set.
cq_replicator_enable	boolean	true	Enables the use of the CQ replicator process.
cq_default_serial_enable	boolean	false	Changes CREATE STREAM defaults to WITHOUT PARTIALS.
cq_node_id	integer	0	Unique identifier for a node in a cluster. Should be unique for each cluster node.
cq_replicator_naptime	number	1	Amount of time in seconds between replicator runs.
cq_run_id	number	null	Sets the run ID for this transaction. If used by the user, care should be taken to ensure the uniqueness per transaction.
cq_multi_runtime_auto_dist_enable	boolean	false	If enabled, distributes serial streams on different runtimes. The runtime is determined using the OID of the first with partials predecessor modulo cq_max_runtimes_processes.

Table 8-3 describes the TruCQ resource allocation options.

Table 8-4 TruCQ Resource Allocation Options

Option	Type	Default	Min/Max	Description
cq_buffer_page_size	number	32 KB	8 KB/ 64 KB	Controls the size of a single CQ buffer page. Data tuples are stored in CQ buffer pages in shared memory to reduce the processing required to copy data tuples between TruCQ processes. This also determines the size of each channel segment. (A channel is an internal communication queue.)
cq_max_channel_segments	number	5	1/20	Controls the maximum number of segments allocated to a channel.
cq_max_runtime_processes	integer	1	1/32	Determines the maximum number of CQ runtime processes to start.
cq_max_archiver_processes	integer	1	1/32	Determines the maximum number of CQ archive processes to start.

Table 8-4 TruCQ Resource Allocation Options (continued)

Option	Type	Default	Min/Max	Description
cq_max_archives	integer	32	1/2048	Specifies the number of channels sent to a relation that is expected to run on the system, for example, regular, checkpoint, or correction channels. TruCQ needs this information because each channel is stored in shared memory so work can be run in parallel. The channels are allocated when the database is started. You should over estimate this parameter to avoid having to restart the database if you add more channels.
cq_planner_shmem_size	number	1 MB	N/A	Controls the size of an internal queue used to communicate query plans inside TruCQ. The 1 MB default is appropriate for most applications.
cq_shared_mem_size	number	16 MB	N/A	Controls the amount of memory allocated for internal communication queues in the TruCQ. If you expect significant data bursts in any stream, you might raise this value.

Allocating Database Memory

To prevent individual users from allocating excessive memory that canot be paged out, UNIX limits the amount of shared memory that a process can allocate. By default, these limits are very small. 32 MB is standard. Because of this limitation, the default TruCQ configuration cannot allocate the required shared memory block needed for high-performance. Instead, the defaults are set so that the CQ engine server can start, even with an untuned kernel that has very low shared memory limitations.

Increasing Shared Memory Limits

To start the server after increasing the TruCQ shared memory parameters, such as cq_shared_mem_size, you will likely need to adjust your OS kernel turning to allow that. The following PostgreSQL documentation provides information on how to adjust or remove this limitation by increasing the approximate kernel tunables.

http://www.postgresql.org/docs/8.3/static/kernel-resources.html

You might need to adjust many of the listed parameters if you must support a large setting for the database max_connections beyond the default of 100. On Linux, you can generally change both SHMMAX (in bytes) and SHMALL (in pages, normally 4096 bytes) to be a large portion, but not all, of the system RAM. 50% of the total RAM is normally large enough to hold even a database that is optimized for heavy memory usage, while, at the same time, not too large to prevent swap allocations through shared memory.

Permanent changes to these settings are made by editing /etc/sysctl.conf as the root user. A script that will produce the appropriate settings to allow up to half of the RAM to be used for shared memory is provided below:

```
#!/bin/bash
mem_bytes='awk '/MemTotal:/ { printf "%0.f",$2 * 1024}' /proc/meminfo' mem_max='expr
$mem_bytes / 2' page_size='getconf PAGE_SIZE' shmall='expr $mem_bytes / $page_size' echo
\# Maximum shared segment size in bytes echo kernel.shmmax = $mem_max echo \# Maximum
number of shared memory segments in pages echo kernel.shmall = $shmall
```

The output from this script consists of lines to add to the sysctl.conf file. Here is an example of its output from a system with 8 GB of RAM:

```
# Maximum shared segment size in bytes kernel.shmmax = 4189255680 # Maximum number of
shared memory segments in pages kernel.shmall = 2045535
```

You could redirect the output from the script directly to /etc/sysctl.conf and then run the sysctl program to get the change to take effect. (This example presumes the above was saved as shm.sh.)

```
./shm.sh >> /etc/sysctl.conf
sysctl -p
```

You do not need to reboot, and the settings will be preserved after restarting.

Increasing Shared Memory Allocation

At startup, the server allocates two blocks of memory in which the bulk of the server operations will normally occur. The shared_buffers parameter is used as a database page caching mechanism for regular database access. Using the cq_shared_mem_size option to increasing it will improve performance on regular (non-streaming) queries.

The parameters controlling these two will default to the following (these lines are in two different spots in the file):

```
shared_buffers = 32MB
#cq_shared_mem_size = 16MB
```

The "#" means that this setting is commented out and is not active. To change it, you must remove the # from the beginning of the line, then set a larger value. An increased one would look like this:

```
cq_shared_mem_size = 256MB
```

One way track the settings that are changed from their defaults is to put them at the end of the postgresql.conf file to keep them together, and commenting out (using #) the locations of settings you do not want applied. The primary setting you will normally need to monitor is shared_buffers. This is normally set to some very small value approximately 100 lines into the file, for example:

```
shared_buffers = 32MB # min 128kB or max_connections*16kB
```

Comment it out will avoid any confusion about which setting takes precedence.

```
#shared_buffers = 32MB # min 128kB or max_connections*16kB
```

Note that on 32-bit platforms, these shared memory parameters will be limited to a maximum of 2 GB even if more RAM is available.

Memory Sizing Guidelines

A configuration snippet from a system with 8 GB of RAM is shown below. You could scale the following up or down depending on how the target system compared to that, keeping the same general ratio:

```
shared_buffers = 512MB
effective_cache_size = 4096MB
cq_shared_mem_size = 512MB
checkpoint_segments = 32
```

The proportions here are:

- shared_buffers—1/16 of RAM

- cq_shared_mem_size—1/16 of RAM
- effective_cache_size—1/2 of RAM
- checkpoint_segments—Isn't a memory parameter. It determines the disk allocation for the database write-ahead logs and is critical for good database performance. 32 allocates approximately1.3GB of disk space for them.

You can increase performance by increasing the two shared memory parameters further, with 1/8 of total RAM still being a fairly conservative value for those. The other CQ-related parameters generally do not need to change on larger or smaller systems. If you have a high number of streams or stream-related objects, you might need to increase the cq_max_channel_segments.

Directory Paths

The main Prime Analytics software is installed into /opt/primeanalytics. Never write anything to this area (by default the files are owned by the root user). Reinstalling Prime Analytics will overwrite everything in this directory.

Changing the Database Location

The main directories used by the Prime Analytics are determined by the settings in the /etc/sysconfig/primeanalytics/primea file. This sets a number of environment variables used by other parts of the software:

export PA_HOME=<root directory>/primeanalytics

export ANT_HOME=$PA_HOME/thirdparty/apache-ant-1.7.1

export JAVA_HOME=$PA_HOME/thirdparty/jdk1.6.0_37

export PATH=$JAVA_HOME/bin:$ANT_HOME/bin:$PA_HOME/TruCQ/bin:$PATH

The database and its related configuration files are stored in a directory tree whose location is identified by the PGDATA environment variable. This variable defaults to /var/opt/primeanalytics/msjor/minor/data. The structure is created by the service primeanalytics initdb command.

You can change the PGDATA directory to point to another location as long as you do this before the initdb step. Your database will live in that alternate location.

The only database requirement is that the database storage tree should be a directory named data. The Linux standard of placing data files into /var is followed, but they can be relocated elsewhere in your file system that makes sense.

Using External or Network Storage for the Database

The Prime Analytics database can be stored on any type of storage as long as it obeys the POSIX fsync mechanics. Avoid the NFS and Linux LVM storage types. Use normal ext3 partitions instead

It is also critical that any disk used does not include any write cache that can be lost, which typically means that only battery-backed write caches are acceptable. See http://www.postgresql.org/docs/current/static/wal-reliability.html for more information on this topic.

Disk Space Estimation

The binaries for the Prime Analytics software take approximately 90 MB to install. You will need that much space in /opt/primeanalytics for each version you want to install simultaneously. The usual way to estimate future disk space is to load a useful subset of the desired data type, measure the table size and extrapolate from there.

User Accounts

PGDATA and CQLOG information is only needed for users who stop and start the database. You do not need to worry about them for regular users. User accounts map into the PostgreSQL role structure by default.

Changing the Prime Analytics HTTP Port

By default, users access the Prime Analytics BI platform server through Port 8080. If you want to change it to a different port, you must manually edit the following files. The examples below change the port to 9080.

- tomcat/conf/server.xml

```
<Connector URIEncoding="UTF-8" port="9080" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="8443" />
```

- biserver-ce/tomcat/webapps/pentaho/WEB-INF/web.xml

```
<context-param>
             <param-name>fully-qualified-server-url</param-name>
             <param-value>http://testarossa5:9080/pentaho/</param-value>
      </context-param>
```

- biserver-ce/tomcat/webapps/pentaho/jsp/Admin.jsp

```
  <div class="btn_manageusers"><a style="color: #0088c2" title="Manage users and
roles" target="_self"
href=<%="http://"+request.getServerName()+":9080/pentaho/AdministrationConsole"%>>Mana
ge Users and Roles</a></div>
```

Configuring and Managing Logs

The standard way to configure the CQ engine is to edit the postgresql.conf file. You can find detailed PostgreSQL configuration information at:

http://www.postgresql.org/docs/8.3/static/runtime-config.html

PostgreSQL error reporting and logging can be found at:

http://www.postgresql.org/docs/8.3/static/runtime-config-logging.html

TruCQ Logging for Problem Resolution

Additional logging that can be helpful if problems occur include:

- Set log_statement to all. This logs every statement executed in by the CQ engine.
- Set the log_min_messages field to DEBUG.
- Set debug_print_plan to true.
- Set log_connections and log_disconnections to true. This logs all connections and disconnections including the various client programs that connect to the database.
- Turn log_hostname to true. This ensures that connection and disconnection messages include resolved hostnames and not IP addresses, which makes debugging easier. The DNS lookup can cause a performance penalty, particularly if not every host connecting is set up properly in DNS.

These suggestions cause a small performance penalty and create larger log files, particularly if you set log_min_messages to DEBUG. While you should usually avoid increasing the log_min_messages volume, doing so can be valuable if an error is easily reproducible.

Relocating the Prime Analytics Logs

Log files are located in three locations. By default, database logs go to the location indicated by the CQLOG environment (set in /etc/sysconfig/primeanalytics/primea). You can change this by editing the postgresql.conf file.

After you turn the logging_collector on, the database does not send normal errors or messages. It redirects them to the file specified in other postgresql.conf sections through the log_directory and log_filename parameters. Under normal conditions, nothing is printed anywhere else.

However, the init script still outputs to $CQLOG. This can be useful if you have problems when starting the server before the postgresql.conf file is read. The error message only appears in the CQLOG location. This commonly occurs when you try to start the database server while it is running. In this case, it cannot open the file where the rotated logs are located because the first server is using them. It will append output to the cq.log file. $CQLOG normally points to the cq.log and indicates startup failed.

If you wanted all log files to go into a log/ directory under the Prime Analytics user's login, you could initially set this up as follows:

```
$ cd ~primeanalytics
$ mkdir -p logs/db logs/app
```

You would then change the CQLOG setting in /etc/sysconfig/primeanalytics/primea:

```
CQLOG=/var/opt/primeanalytics/logs/db/cqstartup.log
```

You might use a rotated log system for the rest of the logs by adding these lines to the postgresql.conf:

```
log_destination = 'stderr' logging_collector = on log_directory =
'/var/opt/primeanalytics/logs/db' log_filename = 'primeanalytics-%a.log'
log_truncate_on_rotation = on
log_rotation_age = 1440 log_rotation_size = 0
```

You can then create a symbolic link in the application's log directory to point to the new structure:

```
$ cd ~primeanalytics/myapp
$ cd apache-tomcat-5.5.23
$ rm -rf logs $ ln -s ~primeanalytics/logs/app logs
```

TruCQ Log Rotation

In addition to the general logging suggestions provided by the http://www.postgresql.org/docs/8.3/static/runtime-config-logging.html, other log rotation options are discussed at http://www.postgresql.org/docs/8.3/static/logfilemaintenance.html

Here is a subset of a postgresql.conf configuration that creates one log file per day and rotates it every week:

```
#------------------------------------------------------------------------------# ERROR
REPORTING AND LOGGING
#------------------------------------------------------------------------------
# -Where to Log
# log_destination = 'stderr'
# Valid values are combinations of
# stderr, csvlog, syslog and eventlog,
# depending on platform. csvlog
# requires logging_collector to be on.
# This is used when logging to stderr: logging_collector = on
# Enable capturing of stderr and csvlog
# into log files. Required to be on for
# csvlogs. # (change requires restart)
# These are only used if logging_collector is on:
# log_directory = 'pg_log'
# directory where log files are written,
# can be absolute or relative to PGDATA log_filename = 'primeanalytics-%a.log'
# log file name pattern,
# can include strftime() escapes log_truncate_on_rotation = on
# If on, an existing log file of the
# same name as the new log file will be
# truncated rather than appended to.
# But such truncation only occurs on
# time-driven rotation, not on restarts # or size-driven rotation. Default is
# off, meaning append to existing files
# in all cases. log_rotation_age = 1440
# Automatic rotation of logfiles will
# happen after that time. 0 to disable. log_rotation_size = 0
# Automatic rotation of logfiles will
# happen after that much log output.
# 0 to disable.
```

This will write the log files to a pg_log directory under the PGDATA subdirectory with the log_directory setting still at the default:

```
#log_directory = 'pg_log'
```

If you want to separate the database logs from the database, change this with an absolute path name instead.

BI Platform Logs

The BI platform log files are located at $PA_HOME/biplatform/tomcat/logs. These include catalina.out, pentaho.log, and truviso.log. The default setting is to show only errors. However you can perform the same modifications as the other logs by editing:

<*BI Server*>/biserver-ce/tomcat/webapps/pentaho/WEB-INF/classes/log4j.xml

Adjusting TruLink Logging

TruLink logging is performed by the Java log4j system. You can change the number of messages you see by editing the customizations/templates/WEB-INF/classes/log4j.properties file.

The best first step to solving problems in this area is to increase the amount of logging done by the application components. Change the customizations/templates/WEB-INF/classes/log4j.properties file to look like this:

```
log4j.logger.com.primeanalytics=DEBUG, primeanalytics
```

In some scenarios, changing the level to TRACE instead of DEBUG might be useful. Note that TRACE is very verbose and likely to impact performance significantly.

TruLink Log Rotation

Because projects use the log4j system, you can adjust log file size and number using the RollingFileAppender class. See http://logging.apache.org/log4j/1.2/apidocs/org/apache/log4j/RollingFileAppender.html for complete details.

Here is a sample log4j configuration designed to rotate these logs based on their size:

```
log4j.appender.primeanalytics=org.apache.log4j.RollingFileAppender
log4j.appender.primeanalytics.File=${catalina.home}/logs/primeanalytics.log
log4j.appender.primeanalytics.maxFileSize=1024MB
log4j.appender.primeanalytics.maxBackupIndex=2
log4j.appender.primeanalytics.layout=org.apache.log4j.PatternLayout
log4j.appender.primeanalytics.layout.ConversionPattern=%d [%t] %-5p %c -%m%n
```

Customizing Server Startup Behavior

The Red Hat chkconfig utility allows you to adjust the run levels that start the server automatically. Here is an example of how to add each Prime Analytics service to chkconfig and to make them start automatically on boot up:

```
/sbin/chkconfig --add primeanalytics
/sbin/chkconfig primeanalytics on
/sbin/chkconfig --add trulink /sbin/chkconfig trulink on
```

The services are expected to be active on are run levels 3 and 5.

Handling RPM Conflicts

Most files installed by Prime Analytics RPMs have a path that includes the version number. The files in /opt/primeanalytics are all uniquely named this way. A few configuration files are shared among all versions:

```
/etc/sysconfig/primeanalytics/primea /etc/init.d/primeanalytics /etc/init.d/trulink
/var/opt/primeanalytics/.bash_profile
```

If you modify any of these files, and try to install a new version, your customized version won't be overwritten. Instead, the new version will be created with the suffix ".rpmnew". You will need to manually merge the contents of the two files. For most version upgrades, you can edit /etc/sysconfig/primeanalytics/primea, The diff utility can help you identify what is customized and what has changed in the new version:

```
cd /etc/sysconfig/primeanalytics diff --side-by-side primeanalytics primeanalytics.rpmnew
```

Using multiple versions concurrently

Because of the configuration file conflicts, RPM will reject any attempt to install two versions of the Prime Analytics software at the same time. However, if you are willing to manage those configuration files by hand, you can have more than one version installed. This makes it easier to roll back to an older version if there proves to be a problem with the newer one.

To install the new version and override the conflict warning, use the -force parameter to RPM:

```
rpm -ivh --force primeanalytics*.rpm
```

You can then switch between similar versions mainly by editing primeanalytics_REL setting in /etc/sysconfig/primeanalytics/primea.

You can even have multiple active startup scripts by copying /etc/init.d/primeanalytics to another name, then creating a new entry in the /etc/sysconfig/primeanalytics directory with that same name. Make sure the port numbers and other shared resources are unique so more than one can work at once.

Backing Up and Restoring Prime Analytics

Backing up and restoring Prime Analytics involves three areas:

- User content—This includes dashboards, reports, charts, and analyses.
- Database repository—This includes historical data used to create the user dashboards, reports, and analyses.
- Configuration files

Note All backup and restore procedures apply to the same Prime Analytics version.

User Data

Content created by Prime Analytics users is stored in the file system solutions repository located at:

$PA_HOME/biplatform/biserver-ce/pentaho-solutions

User-created OLAP and reporting models are located in the following directories:

./primeanalytics/biplatform/biserver-ce/pentaho-solutions/admin/resources/metadata
./primeanalytics/biplatform/biserver-ce/pentaho-solutions/system/olap/datasources.xml

Database Repository

The Prime Analytics BI platform contains three databases:

- Hibernate
- Quartz
- Connector metadata

The hibernate database stores BI platform system configuration such as users, user roles, and user settings. The quartz database stores BI platform scheduling information. The connector database stores realtime SQL information used for creating dashboards. The quartz, hibernate, and connector databases are created on TruCQ and are included with the BI platform.

Prime Analytics scripts back up and restore the BI platform and the TruCQ database. These scripts are located in the following directories:

- TruCQ database—$PA_HOME/bin/backup_restore/Database
- BI Platform—$PA_HOME/bin/backup_restore/BIPlatform

The following practices ensure the safety and efficiency of the backup and restore process:

- Stopping the BI platform server before you back up the BI platform is recommended to avoid any transient issues.
- The backup.dir property in backup_restore.properties (available in both the BIPlatform and Database directories) should point to the same shared network directory with write permissions for both the primea and bipuser users.
- Back up the shared network drive periodically and verify a mechanism exists to recover the data.
- Backing up the BI platform and database using CRON jobs during non-peak hours is recommended to minimize the network impact.

The following topics provide details and prerequisites for backing up and restoring Prime Analytics:

- Backing Up the BI Platform, page 8-27
- Backing Up the Database, page 8-28
- Restoring the BI Platform, page 8-28
- Restoring the Database, page 8-29

Backing Up the BI Platform

Before you back up the BI platform, verify that:

- The backup directory is specified in the backup_restore.properties file. that is, backup.dir.
- The bipuser user has write permission to the backup directory specified in backup_restore.properties.

Note The BI platform backup file name is biplatform-TIMESTAMP.tar.gz , where TIMESTAMP is the backup date and time. Do not change the backup file name.

To back up the BI platform:

Step 1 Log into the BI platform as the bipuser.

Step 2 Change to the BI platform backup directory:

```
cd $PA_HOME/bin/backup_restore/BIPlatform
```

Step 3 Start the backup:

```
./backup_biplatform.sh
```

Backing Up the Database

Before you back up the database, verify the following:

- Specify the backup directory in the backup_restore.properties file.
- Verify the user running the database backup script has write permission to the backup directory specified in the backup_restore.properties file.

Note The Prime Analytics database backup does not include TruCQ related configurations, for example, pg_hba.conf, pg_ident.conf, and postgresql.conf located in /var/opt/primea/data/. If a database configuration, such as a TruCQ port, was changed by editing the file, restore will not include the customized port. Database backup also does not back up the application or project directory created by users. You will need to back up these directories using another backup method.

To back up the BI platform:

Step 1 Log into the Prime Analytics TruCQ server as the primea user. If you are switching users, use the su - primea command.

Step 2 Change to the database backup directory:

```
cd $PA_HOME/bin/backup_restore/Database
```

Step 3 Start the backup:

```
./backup_db_repository.sh
```

Restoring the BI Platform

The BI platform restore stops the BI server before performing a restore, so you should advise users the server will not be available during the restore.

To restore the BI platform:

Step 1 Log into the BI platform server as the bipuser.

Step 2 Change to the BI platform restore directory:

```
cd $PA_HOME/bin/backup_restore/BIPlatform
```

Step 3 Start the restore:

```
restore_biplatform.sh <BACKUP_FILE_LOCATION>
```

Where BACKUP_FILE_LOCATION is the tar.gz file with location.

Note Prime Analytics validates the backup file to ensure that it was created by Prime Analytics. If not, the restore will not proceed.

Step 4 After the restore is complete, in the Prime Analytics home page, click the **Refresh** tool above the navigation tree to refresh the GUI display.

Restoring the Database

Before you begin the database restore, verify that the primea user has write permission to the /tmp directory. During the restore, you will stop the BI platform and database servers.

If you are restoring the TruCQ database in an HA configuration where the BIDB (BI platform repository) and TruCQ databases are on different servers, make sure to restore the backup files correctly.

- Restore files starting with bisample-<time>.tar.gz to the TruCQ database
- Restore files starting with repository-<time>.tar.gz to the BIDB database

To restore the Prime Analytics database:

Step 1 Log into the Prime Analytics database (TruCQ) server as the root.

Step 2 Stop the BI platform:

service biplatform stop

Step 3 Stop the TruCQ server:

For TruCQ database:

service trucq stop

For BIDB database (HA):

service bitrucq stop

Step 4 Change to the database restore directory:

```
cd $PA_HOME/bin/backup_restore/Database
```

Step 5 Start the database restore:

./restore_db_repository.sh <BACKUP_FILE_LOCATION>

Where BACKUP_FILE_LOCATION is tar.gz file with location.

Note Prime Analytics validates the backup file to ensure that it was created by Prime Analytics. If not, the restore will not proceed.

Step 6 If you restored the database to a new environment, such as a new hostname or IP address (assuming the BI platform is still the same) manually update the following files with the new values:

- $PA_HOME/biserver-ce /pentaho-solutions/system/applicationContext-spring-security-hibernate.properties
- $PA_HOME/biserver-ce/tomcat/webapps/pentaho/META-INF/context.xml
- $PA_HOME/biserver-ce /pentaho-solutions/system/hibernate/postgresql.hibernate.cfg.xml
- $PA_HOME/biserver-ce /pentaho-solutions/system/quartz/quartz.properties

You might also need to change the entries in the TVDB database because it stores the TruCQ engine details along with server name.

Step 7 After the restore is complete, start the TruCQ database server:

For TruCQ database:

service trucq start

For BIDB database (HA):

service bitrucq start

Step 8 Validate the restore database and delete the original data folder from /tmp.

Step 9 Start the BI platform:

service biplatform start

Starting and Stopping Prime Analytics Servers

Table 8-5 provides commands that you can use to start and stop the Prime Analytics BI platform and TruCQ engine, You must run these commands as the root user.

In general, start and stop the BI platform using the service biplatform command as the root. Alternatively, you can start and stop the BI platform using ./start-primebip.sh and ./stop-primebip.sh as the root.

Table 8-5 Prime Analytics Server Commands

Action	Command
Stop the BI platform.	service biplatform stop
Start the BI platform	service biplatform start
Stop the TruCQ engine	service trucq stop
Start the TruCQ engine	service trucq start