



Pentaho Business Analytics Administrator's Guide



This document supports Pentaho Business Analytics Suite 4.8 GA and Pentaho Data Integration 4.4 GA, documentation revision October 31, 2012.

This document is copyright © 2012 Pentaho Corporation. No part may be reprinted without written permission from Pentaho Corporation. All trademarks are the property of their respective owners.

Help and Support Resources

If you have questions that are not covered in this guide, or if you would like to report errors in the documentation, please contact your Pentaho technical support representative.

Support-related questions should be submitted through the Pentaho Customer Support Portal at <http://support.pentaho.com>.

For information about how to purchase support or enable an additional named support contact, please contact your sales representative, or send an email to sales@pentaho.com.

For information about instructor-led training on the topics covered in this guide, visit <http://www.pentaho.com/training>.

Limits of Liability and Disclaimer of Warranty

The author(s) of this document have used their best efforts in preparing the content and the programs contained in it. These efforts include the development, research, and testing of the theories and programs to determine their effectiveness. The author and publisher make no warranty of any kind, express or implied, with regard to these programs or the documentation contained in this book.

The author(s) and Pentaho shall not be liable in the event of incidental or consequential damages in connection with, or arising out of, the furnishing, performance, or use of the programs, associated instructions, and/or claims.

Trademarks

Pentaho (TM) and the Pentaho logo are registered trademarks of Pentaho Corporation. All other trademarks are the property of their respective owners. Trademarked names may appear throughout this document. Rather than list the names and entities that own the trademarks or insert a trademark symbol with each mention of the trademarked name, Pentaho states that it is using the names for editorial purposes only and to the benefit of the trademark owner, with no intention of infringing upon that trademark.

Company Information

Pentaho Corporation
Citadel International, Suite 340
5950 Hazeltine National Drive
Orlando, FL 32822
Phone: +1 407 812-OPEN (6736)
Fax: +1 407 517-4575
<http://www.pentaho.com>

E-mail: communityconnection@pentaho.com

Sales Inquiries: sales@pentaho.com

Documentation Suggestions: documentation@pentaho.com

Sign-up for our newsletter: <http://community.pentaho.com/newsletter/>

Contents

Introduction	5
Prerequisites	5
Accessing the Pentaho Enterprise Console	5
Service Control	6
Master Service Control Scripts From the Graphical Installer	6
Individual Service Control Scripts	6
Starting the BA Server At Boot Time On Linux	6
Starting the BA Server At Boot Time On Solaris	7
Starting the BA Server At Boot Time On Windows	7
Pentaho Enterprise Console Configuration	9
Install or Update License Keys	10
Installing or Updating License Keys Using the Pentaho Enterprise Console	10
Configuring the Proxy Trusting Filter	11
Changing the Admin Credentials for the Pentaho Enterprise Console	11
Cleanup and Audit Reports	12
More About Audit Reports	13
The Data Integration Server	14
Configuring the Enterprise Console	14
Connecting to the Data Integration Server	15
Establishing Data Sources	16
Creating JNDI Data Connections	17
Adding a JNDI Data Connection to JBoss	17
Adding a JNDI Data Connection to Tomcat	18
Adding a Simple JNDI Data Source For Design Tools	19
Creating JDBC Database Connections	19
Adding a JDBC Driver	20
Adding a JDBC Database Connection In Enterprise Console	21
Using Advanced JNDI Connection Options in the Enterprise Console	21
Adding a Data Source in the Pentaho User Console	21
Adjusting Name Restrictions in Data Source Wizard	28
Restricting Metadata Models to Specific Client Tools	28
Managing Schedules	30
Entering Schedules in the Schedule Creator Dialog Box	30
Associating Schedules with Action Sequences	30
Removing Action Sequence Files from a List	31
Examining the List of Schedules	31
Configuring the BA Server	33
Software Updates	33
Email Settings	33
Specifying Your Publisher Password	34
Specifying Global and Session Variables	34
Changing the Location of the Server Log File	34
Changing the Tomcat Port	34
Hiding Report Parameters In URLs	35
Setting the Geo Map Rendering Wait Time	35
Adding Geo Map Support to Data Source Wizard	35
Dashboard Printing in a Headless Linux Environment	37
Enabling Dashboard Printing on Headless CentOS and Red Hat	37
Enabling Dashboard Printing on Headless Ubuntu	37
Solution Repository and Cache Management	38
Backing Up the Solution Repository	38
Checking the Status of the Solution Repository	38
Configuring the Solution Repository	38
Hibernate Dialects	39
Refreshing the Reporting Data Cache	40

Clearing the Content Repository.....	40
Clearing the Mondrian Cache.....	40
Content Promotion (Lifecycle Management).....	41
Lifecycle Management Checklist.....	41
Preparing the Configuration for Promotion.....	42
Promoting Solutions.....	42
Migrating Data Sources.....	43
Promoting Schedules.....	43
Artifact Cleanup.....	44
Logging, Performance Monitoring, and Troubleshooting.....	45
Viewing Errors.....	45
Viewing Environment Settings.....	45
Viewing Security Settings.....	45
Configuring and Testing System Settings.....	45
Monitoring Performance.....	46
Changing the Show for Last Interval.....	47
Refreshing the Metrics Values.....	47
Preventing Timeout Transaction Errors Associated with Scheduled Reports.....	48
Using Performance Charts.....	48
Estimating the Size of the Audit Log.....	50
Defining Result Row Limit and Timeout.....	51
Column Metadata Security Restrictions On Columns With Inner Joins.....	52
Disabling Server and Session-Related Timeouts in the Pentaho User Console.....	52
Log Rotation.....	52
Troubleshooting.....	53
Address Already In Use: JVM_Bind When Starting Enterprise Console.....	54
Case-Insensitivity for Usernames, Passwords, and Filenames.....	54
HTTP 500 or "Unable to connect to BA Server" Errors When Trying to Access Enterprise Console.....	54
Windows Domains Won't Authenticate When Using the JTDS Driver.....	54
Geo Maps partially rendered.....	54
Getting Support.....	55
Working From the Command Line Interface.....	56
Managing Enterprise Edition Keys (CLI).....	56
Resetting the Default Access Control Lists (CLI).....	56

Introduction

This guide contains instructions for post-install configuration and maintenance of Pentaho software.

Prerequisites

Console features and functions are completely dependent on the BA Server. You must have a BA Server up and running prior to starting your Pentaho Enterprise Console.

Accessing the Pentaho Enterprise Console

To access the console for the first time you must authenticate using default credentials:

1. Start the Pentaho Enterprise Console as described.
2. Go to **http://localhost:8088/** in your browser.
3. Enter the default user name, **admin**.
4. Enter the default password, **password**.
5. Click **OK**
The console home page appears.

Service Control

This section contains instructions for starting and stopping the BA Server, DI Server, and Enterprise Console.

Master Service Control Scripts From the Graphical Installer

If you installed Pentaho Business Analytics on a single machine through Pentaho's graphical installation utility, then there is a global service control script in the top-level directory that you installed to: **ctlscript** (.sh on Linux, .bat on Windows). This will start the BA Server, DI Server, Enterprise Console, solution database, and hsqldb sample database. **ctlscript** takes the following arguments:

- start
- stop
- restart
- status
- help

Individual Service Control Scripts

If you installed the BA Server or DI Server through archive packages, there are individual control scripts on your system for each service (replace the .sh with a .bat if you are on Windows):

BA Server: /pentaho/server/biserver-ee/**start-pentaho.sh** and **stop-pentaho.sh**

DI Server: /pentaho/server/data-integration-server/**start-pentaho.sh** and **stop-pentaho.sh**

Enterprise Console: /pentaho/server/enterprise-console/**start-pec.sh** and **stop-pec.sh**

hsqldb sample data: /pentaho/server/hsql-sample-database/**start_hypersonic.sh** **stop_hypersonic.sh**


Alternatively, you could interact directly with the provided Tomcat and Jetty servers through their standard start and stop scripts. However, you will have to examine the Pentaho-provided scripts listed above to get the environment settings from them, then transfer those settings to your global system configuration. Without the proper environment settings, Pentaho services will not work correctly.

Starting the BA Server At Boot Time On Linux

This procedure assumes that you will be running your BA Server and Pentaho Enterprise Console server under the **pentaho** local user account, as recommended by Pentaho and explained earlier in this guide. If you are using a different account to start these services, use it in place of the pentaho user account in the script below.

You can start and stop the BA Server at any time by running the **start-pentaho.sh** and **stop-pentaho.sh** scripts. To start the Tomcat server automatically at boot time, and stop automatically during shutdown, follow the below procedure.

1. With root permissions, create a file in /etc/init.d/ called **pentaho**.
2. Using a text editor, copy the following content into the new pentaho script, changing **postgresql** to the name of the init script for your database if it is running on the remote machine, or remove **postgresql** entirely if you are using a remote database. Secondly, you must adjust the paths to the BA Server and Pentaho Enterprise Console scripts to match your situation.

 **Note:** The below script was tested on Red Hat Enterprise Linux. You may have to modify the details of the script if you use different distributions or other Unix-like operating systems, different shells, or different init systems.

```
#!/bin/sh
### BEGIN INIT INFO
# Provides: start-pentaho stop-pentaho
# Required-Start: networking postgresql
# Required-Stop: postgresql
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
```

```
# Description: Pentaho BA Server
### END INIT INFO

case "$1" in
"start")
su - pentaho -c "/home/pentaho/pentaho/server/biserver-ee/start-pentaho.sh"
su - pentaho -c "cd /home/pentaho/pentaho/server/enterprise-console && ./start-
pec.sh"
;;
"stop")
su - pentaho -c "/home/pentaho/pentaho/server/biserver-ee/stop-pentaho.sh"
su - pentaho -c "cd /home/pentaho/pentaho/server/enterprise-console && ./stop-
pec.sh"
;;
*)
echo "Usage: $0 { start | stop }"
;;
esac
exit 0
```

3. Save the file, then open `/home/pentaho/pentaho/server/biserver-ee/start-pentaho.sh`.
4. Change the last **if** statement to match the following example:

```
if [ "$?" = 0 ]; then
cd "$DIR/tomcat/bin"
export CATALINA_OPTS="-Xms256m -Xmx768m -XX:MaxPermSize=256m -
Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000"
env JAVA_HOME=$_PENTAHO_JAVA_HOME sh ./startup.sh
fi
```

5. Save the file and close the text editor.
6. Make the init script executable.

```
chmod +x /etc/init.d/pentaho
```

7. Add the pentaho init script to the standard runlevels so that it will run when the system starts, and stop when the system is shut down or rebooted, by using the `update-rc.d` command.

This command may not exist on your computer if it is not Debian-based. If that is the case, consult your distribution documentation or contact your distribution's support department to determine how to add init scripts to the default runlevels.

```
update-rc.d pentaho defaults
```

The Pentaho BA Server will now start at boot time, and shut down when the system stops or restarts.

Starting the BA Server At Boot Time On Solaris

Solaris 10 uses two methods to configure system services: traditional RC scripts (the "legacy" method), and `svcadm`, which is the newer, preferred method. You should create a startup script or service definition file according to your company standards, and add it to the appropriate runlevel. Pentaho does not provide a reference startup script or service configuration file for Solaris at this time.

Starting the BA Server At Boot Time On Windows

You must have the Windows Resource Kit installed in order to proceed.

To start the Tomcat and Pentaho Enterprise Console servers automatically at boot time, and stop automatically during shutdown, follow the below procedure.



Note: The paths in the example commands may have to be changed to match your environment.

1. Open a Command Prompt window.

2. Run the **service.bat** script provided by Apache for Tomcat service configuration.

```
C:\Documents and Settings\pentaho\pentaho\server\biserver-ee\tomcat\bin\service.bat
install
```

3. Close all programs and restart Windows.
4. Log in as the user account that will control the BA Server.
5. Navigate to the `\pentaho\server\biserver-ee\tomcat\bin\` directory and run the **tomcat6w.exe** program there.
The Apache Tomcat service configuration dialogue will appear.
6. In the **General** tab, change the **Startup type** to **Automatic**.
7. If you did not log on as the user that will control the BA Server as a service, then in the **Log on** tab, enter the proper Windows credentials for that user.
8. In the **Java** tab, set **Initial memory** to **768** (or whatever your preferred BA Server Java Xms setting is).
9. In the **Java** tab, set **Maximum memory pool** to **768** (or whatever your preferred BA Server Java Xmx setting is).
10. In the **Java** tab, add an absolute path to your **pentaho** directory (or whatever the parent or grandparent directory of **biserver-ee** is) in a **-Dpentaho.installed.licenses.file** parameter, followed by the **.installedLicenses.xml** file name, as in the following example:

```
-Dpentaho.installed.licenses.file="C:\Documents and Settings\pentaho\pentaho
\installedLicenses.xml"
```

This establishes a path to the file that stores your Pentaho licenses. If you have already installed licenses for this BA Server instance, setting this variable correctly should prevent any licensing issues from occurring. You may have to search your system for the **.installedLicenses.xml** file and change the path appropriately if you have already installed Pentaho licenses and performing this step does not work on your first attempt. If you have not yet installed any licenses, this step is not necessary because the BA Server will look in several logical locations relative to the Tomcat directory.

11. Ensure that your Pentaho solution database is configured to start and run as a service in Windows.

The BA Server will not start if the solution database is not running.

12. Restart Windows and ensure that the BA Server starts automatically, is available, and has proper access to any license files that you may have installed prior to executing this procedure.

The Pentaho BA Server will now start at boot time.



Note: Enterprise Console uses a separate application server and will not start at boot time through this process. Enterprise Console will have to be manually started through the **start-pec.bat** script when needed.

Pentaho Enterprise Console Configuration

Before you use the Pentaho Enterprise Console to manage the Data Integration Server, BA Server, or both, you must install license keys (referred to in the Enterprise Console interface as **subscription files**) and configure the console to locate certain files and directories. The first time you access the Pentaho Enterprise Console, the setup page will appear. If no license keys have been installed, all you will see is a license key dialog box.

Note:



See [The Data Integration Server](#) for more information.

The Pentaho Enterprise Console displays this page any time the configuration is not correct, for example:

- The console configuration has been manually altered and is no longer valid
- The BA Server installation location has moved or been renamed
- The Enterprise Edition key has expired or is invalid

Accessing the Configuration Page

To access the configuration page at any time click the setup icon (a wrench and a flat-head screwdriver) on the toolbar in the Pentaho Enterprise Console. The console setup page contains a list of Enterprise Edition license keys, plus the following settings:

Setting	Description
Solution Directory	Paste the path to the solutions directory of the BA Server you want to administer into the Solution Directory text box; for example, /pentaho/server/biserver-ee/pentaho-solutions/.
Backup Directory	Paste the path to your backup directory in the Backup Directory text box.
Pentaho Web-App Path	Paste the path to the Web application directory of the BA Server you want to administer into the Pentaho Web-App Path text box, for example: /pentaho/server/biserver-ee/tomcat/webapps/pentaho/

Setting	Description
Platform Administrator User Name	Enter the name of the administrator of the BA platform in the Platform Administrator User Name text box.
Platform Server Status Check Period (milliseconds)	Enter the status check period value in the Platform Server Status Check Period text box. This is the time period that the console waits before pinging the BA Platform server for server status. The default value is provided but you can change it.



Note: For the Pentaho Enterprise Console to function correctly on a minimal basis, the solution directory and a valid Enterprise Edition key must be found.

Install or Update License Keys

License keys enable your users to run Pentaho products. Your Pentaho Welcome Kit contains license keys, which are stored in LIC files. Each Pentaho product has its own LIC file. If you did not receive your welcome kit, or if you have lost the LIC files, contact your Pentaho support representative. If you do not yet have a support representative, contact your Pentaho salesperson. You activate a license key by installing an LIC file from the user accounts that will run the related Pentaho product. If you start the BA or DI Server automatically at boot time, install the LIC files from the user account that is responsible for system services. You can install LIC files from multiple local user accounts. These settings depend on the operating system and the system configurations. If you need more information about user accounts for your operating system, see the documentation for your operating system.

Whether you are activating license keys for the first time or updating expired keys, there are two ways to install LIC files from user accounts. Pentaho supplies the Pentaho Enterprise Console (PEC) that, among other things, enables you to access user accounts and install LIC files.

If you performed a manual installation, you may have installed PEC on a different machine than the BA or DI Server. If this is your situation, you must install the LIC files from the command line. However, if you run the BA or DI Server on the same machine that runs PEC, you can use it to install LIC files. This configuration is the default if you performed a graphical or archive installation.

Installing or Updating License Keys Using the Pentaho Enterprise Console

1. Copy your LIC files to the server that runs the Pentaho Enterprise Console.
2. Log into the Pentaho Enterprise Console by opening a web browser and navigating to `www.<serverHostName>:8088`, changing `<serverHostName>` to the host name or IP address of your BA or DI server.
3. Click the + (plus) button in the upper-right corner of the **Subscriptions** section. **Browse**, then navigate to your LIC files and open them.
The Install License dialog box appears.
4. Click Browse and navigate to the LIC files you want to install and open them.
5. After opening all LIC files, click **OK**.

You can configure your licensed products through the Pentaho Enterprise Console.

Installing License Keys from the Command-line Interface

1. Download the related archive package for the Pentaho product LIC file you want to install.
2. Copy your LIC files to the server that runs the Pentaho Enterprise Console.
3. Navigate to the `/pentaho/server/enterprise-console/license-installer/` directory, or the `license-installer/` directory that was part of the archive package you downloaded.
4. Run the license installation script.
 - a) **For Linux:** Run `install_license.sh` with the install switch and the location and name of your LIC file as a parameter. You can specify multiple LIC files separated by spaces. Be sure to use backslashes to escape any spaces in the path or file name.
`install_license.sh install /home/pgibbons/downloads/Pentaho\ BI\ Platform\ Enterprise\ Edition.lic`

- b) **For Windows:** Run `install_license.bat` with the `install` switch and the location and name of your license file as a parameter.
- ```
install_license.bat install "C:\Users\pgibbons\Downloads\Pentaho BA Platform Enterprise Edition.lic"
```

## Configuring the Proxy Trusting Filter

If you have set the BA Server to run on a specific IP address or hostname other than the default 127.0.0.1, you must modify the trusted proxy between the BA Server and Pentaho Enterprise Console to match that address or hostname.

1. Stop the Pentaho Enterprise Console.
2. Stop the BA Server.
3. Open `biserver-ee/tomcat/webapps/pentaho/WEB-INF/web.xml` and search for **TrustedIpAddr**s.



**Note:** The `param-value` immediately below **TrustedIpAddr**s is a comma-separated list of IP addresses that should be trusted.

```
<filter>
 <filter-name>Proxy Trusting Filter</filter-name>
 <filter-class>org.pentaho.platform.web.http.filters.ProxyTrustingFilter</filter-
class>
 <init-param>
 <param-name>TrustedIpAddr</param-name>
 <param-value>127.0.0.1</param-value>
 <description>Comma separated list of IP addresses of a trusted hosts.</
description>
 </init-param>
</filter>
```

4. Add the IP address of the host running the Pentaho Enterprise Console.
5. Open `enterprise-console/resource/config/console.xml` and search for **platform-username**.
6. Change the default **admin** user name, (default is joe) to your desired admin user name.
7. Start the BA Server.
8. Start the Pentaho Enterprise Console.

## Changing the Admin Credentials for the Pentaho Enterprise Console

The default user name and password for the Pentaho Enterprise Console are **admin** and **password**, respectively. You must change these credentials if you are deploying the BA Server to a production environment. Follow the instructions below to change the credentials:

1. Stop the Pentaho Enterprise Console.

```
/pentaho/server/enterprise-console/stop-pec.sh
```

2. Open a terminal or command prompt window and navigate to the `/pentaho/server/enterprise-console/` directory.
3. Execute the **pec-passwd** script with two parameters: the Enterprise Console username and the new password you want to set.

This script will make some configuration changes for you, and output an obfuscated password hash to the terminal.

```
./pec-passwd.sh admin newpass
```

4. Copy the hash output to the clipboard, text buffer, or to a temporary text file.
5. Edit the `/pentaho/server/enterprise-console/resource/config/login.properties` file with a text editor.
6. Replace the existing hash string with the new one, leaving all other information in the file intact.

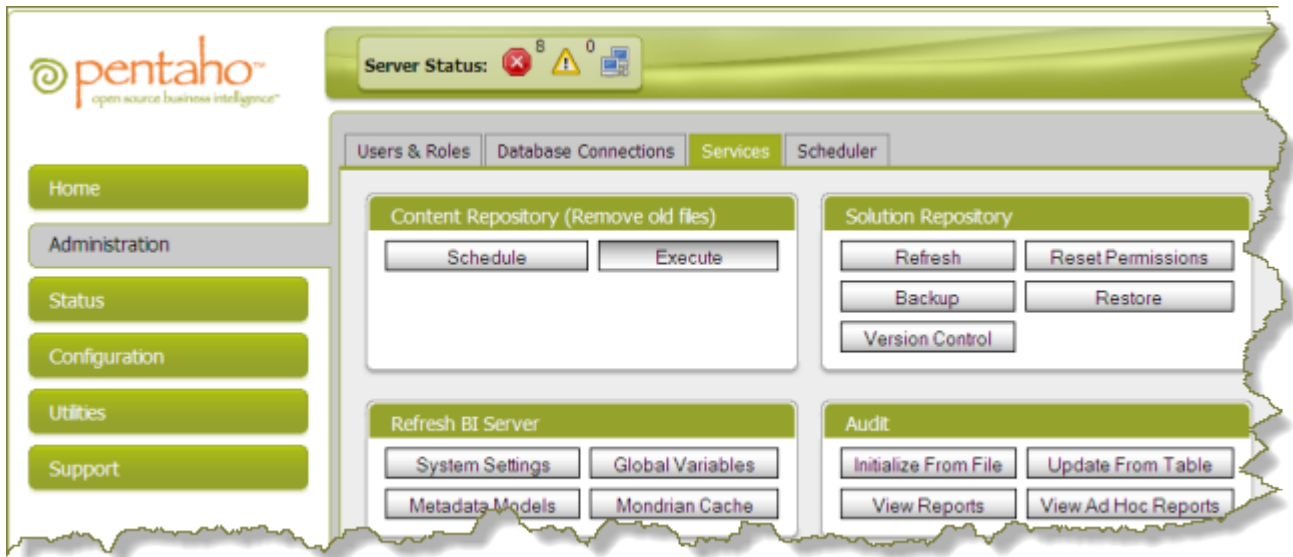
```
admin: OBF:1uo91vn61ymf1yt41v1plym71v2plytilylzl1vnwlunp,server-
administrator,content-administrator,admin
```

7. Save and close the file, and restart Enterprise Console.

The Pentaho Enterprise Console password is now changed.

## Cleanup and Audit Reports

The **Services** page associated with the **Administration** menu item provides you with the ability to manage system-level services. Many of the options on this page are associated with managing services related to the solution and content repositories; for example, clearing old data from the content repository and restoring the solution repository. Also included are tools that allow you to audit usage of data in the repository; for example, auditing helps you determine what type of repository-related data (such as reports) is generating the highest interest among users. The table below provides you with a short description of each option on the Services page:



Option	Description
<b>Schedule</b>	Schedules the daily removal of files created in the content repository located at <code>/pentaho-solutions/system/content/</code> , which are over 180 days old. To change, the number of days, edit the solution file <b>clean_repository.xaction</b> located in <code>/pentaho-solution/admin</code> . To change the recurrence, edit the solution file, <b>schedule-clean.xaction</b> located in <code>/pentaho-solution/admin</code>
<b>Execute</b>	Removes files created in the content repository located at <code>/pentaho-solutions/system/content/</code> that are over 180 days old. To change, the number of days, edit the solution file, <b>clean_repository.xaction</b> located in <code>/pentaho-solutions/admin/</code> .
<b>Refresh</b>	Synchronizes the RDBMS-based solution repository with the file system when solution files are manually added or edited
<b>Backup</b>	Backs up ACLs to <code>pentaho.xml</code> , zips <code>pentaho-solutions</code> and places it in the configured backup directory (see Note)
<b>Version Control</b>	Updates the solution files on the local file system from a version control system (SVN/CVS). The link executes a script called <b>versionControl.bat</b> or <b>versionControl.sh</b> found in the solution repository. The script commands get an update from the version control system into the local file system.

Option	Description
<b>Reset Permissions</b>	Deletes all the solution files and their permissions from the RDBMS-based solution repository and copies the files with default permissions from the master solution repository on the local file system.
<b>Restore</b>	Restores system from a backup list (restores ACLs and/or solutions) (see Note)
<b>System Settings</b>	Reloads all the configurations files found in pentaho-solutions/System.
<b>Global Variables</b>	Updates global variables by re-executing registered solution files.
<b>Metadata Models</b>	Refreshes the Metadata cache when models are added, edited, or deleted in the solution repository
<b>Mondrian Cache</b>	Clears the Mondrian schema and data cache.
<b>Initialize from File</b>	Loads the audit log file into a database used for Audit Reports and Ad-hoc Audit Reports when using the file-based solution repository.
<b>Update from Table</b>	Loads the audit log table into a database used for Audit Reports and Ad-hoc Audit Reports when using the RDBMS-based solution. repository,
<b>View Reports</b>	Provides a list of over 20 pre-built reports for analyzing report usage, user access, and report processing time.
<b>View Ad Hoc Reports</b>	Provides an interface for administrators to create their own queries of the audit information.

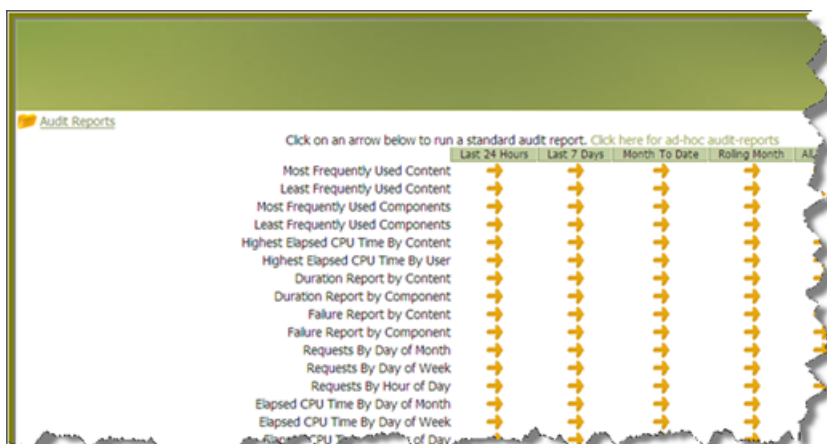


**Note:** The **Backup** and **Restore** options do not commit until progress is 100%; therefore, canceling is non-destructive. Resetting ACLs returns to the last restore (or default).

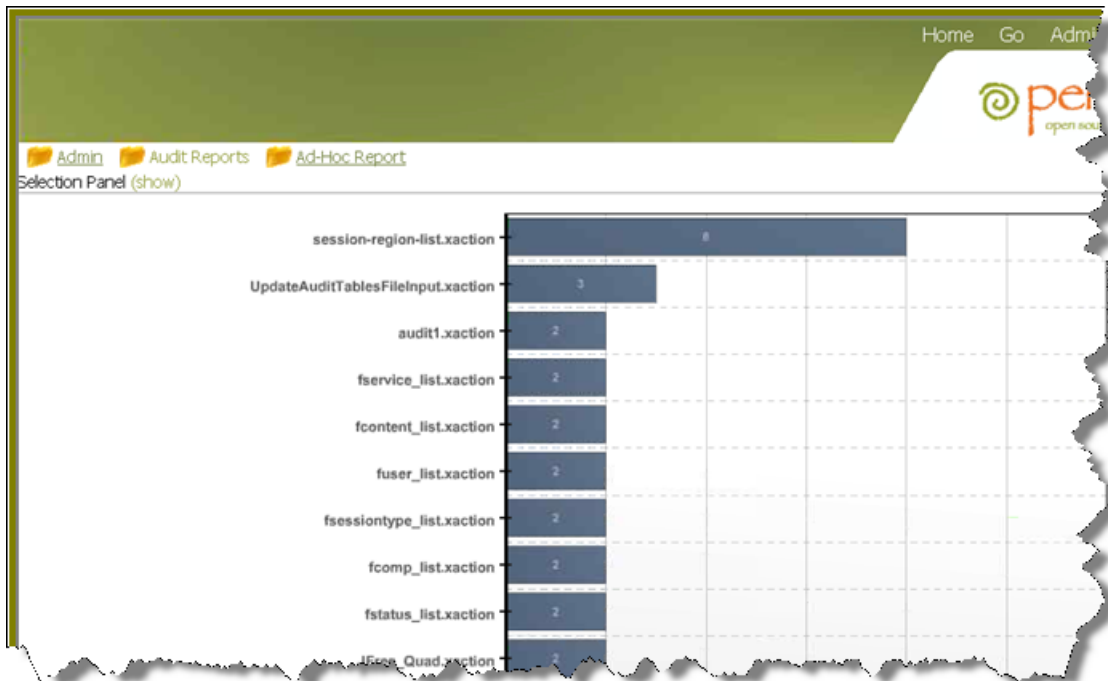
## More About Audit Reports

Audit reports help you understand and optimize your business intelligence applications. Audit reports provide immediate insight into user activity, system performance, most popular reports, and more. Audit reports can also be useful for tracking compliance and ensuring that resources are used according to company policy. Beyond that, understanding system performance and most popular reports allows you to progressively tune your deployment, and ensure that business users have the access to the most critical information necessary for doing their jobs.

In the sample below, the administrator has clicked **View Reports** under **Services** in the Pentaho Enterprise Console to open the Audit Reports feature in the Pentaho User Console:



The administrator then clicks first arrow next to **Most Frequently Used Content** to display the .xaction (solution) files that are most frequently being accessed by users in the last 24 hours:



The administrator can continue to drill down to obtain usage information for the previous week, month-to-date, all time, and more.

### SQL-Based Versus File-Based Audit Logging

Out of the box, the BA Server is configured to log all audit entries to a SQL table. SQL-based logging offers the advantage of incremental updates to the audit reporting tables. The BA Server can also be configured to log audit entries to a text file. The file-based audit logger streams all audit information to a text file.

The type of audit logging you select depends on the requirements of your environment. High-use production environments result in a significant number of SQL inserts that could impact server performance. In this case, you may choose to switch to file-based audit logging. See [Switching to File-Based Audit Logging](#) for instructions.

## The Data Integration Server

Pentaho Business Analytics installation provides you with a complete set of Business Intelligence tools, including the Data Integration server, a dedicated ETL server.

Among other things, the Data Integration server...

- Executes ETL jobs and transformations using the Pentaho Data Integration engine
- Allows you to manage users and roles (default security) or integrate security to your existing security provider such as LDAP or Active Directory
- Includes content management providing the ability to centrally store and manage your ETL jobs and transformations which encompasses full revision history on content and features such as sharing and locking for collaborative development environments
- Allows you to schedule activities and monitor scheduled activities on the Data Integration server from within designer environment

### Configuring the Enterprise Console

The Data Integration Server is an independent, separate server from the Pentaho BA Server. The Pentaho Enterprise Console can be used to manage the BA Server, the Data Integration server, or both. If you are setting up a dedicated server for ETL, follow the instructions below to configure the Pentaho Enterprise Console to display administrative functions associated with Pentaho Data Integration exclusively.

1. Click the Setup button in the upper-right corner of the Pentaho Enterprise Console home page.

The Pentaho Enterprise Setup page appears.


2. Enable **Running Data Integration Server Only**.
3. Click **OK** to exit the setup page.

The Pentaho Enterprise Console displays a page that contains information associated with Pentaho Data Integration. For more information, see *Getting Started with Pentaho Data Integration* available in the Pentaho InfoCenter.

## Connecting to the Data Integration Server

You must register your Data Integration server to link it to the Pentaho Enterprise Console. After registration you can monitor activity associated with jobs and transformations being processed by the Data Integration server from the PDI dashboard in the Pentaho Enterprise Console.

To register your Data Integration server:

1. Make sure your Data Integration server is up and running.
2. In the Pentaho Enterprise Console home page, click **Pentaho Data Integration**.
3. Click  to open the **Carte Configuration** dialog box.
4. In the **Carte Configuration** dialog box, enter the Data Integration server URL, (for example, `http://localhost:9080/pentaho-di/kettle`), the server user name, and password.



**Note:** You can still configure the Pentaho Enterprise Console to connect to a basic Carte server using the appropriate URL; (for example, `http://localhost:80/kettle/`).

5. Click **OK** to complete the registration.  
A message appears if the connection to the Data Integration server is successful. If you see an error message, check to make sure your Data Integration server is running and that your access credentials are correct.

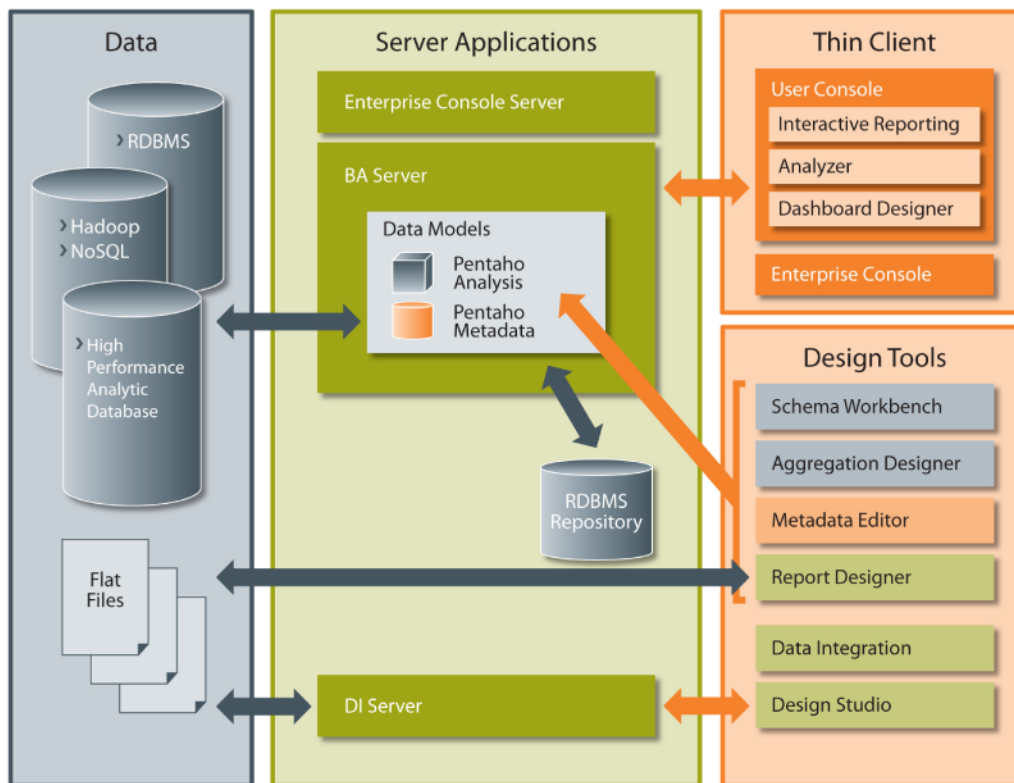


# Establishing Data Sources

The BA Server can work with these types of data sources:

- RDBMS, Relational OLTP or OLAP
- Big Data (Hadoop, Cassandra, Mongo DB)
- Flat files (semi-structured, XML, Excel)
- High Performance Analytical Database
- Other - Web Service, RSS Feed, Google Analytics, etc.

The below image shows how the data is processed through Pentaho Business Analytics so that business users can more easily use their data to make informed business decisions.



	Dashboard Designer	Analyzer	Report Designer	Data Integration
<b>Data source</b>			X	X
<b>Metadata</b>	X		X	X
<b>Schema</b>		X	X	X
<b>Straight JDBC</b>			X	X
<b>NoSQL</b>			X	X
<b>Other</b>			X*	X

\*PRD can connect to any data source by connecting via PDI.

## RDMBS

A Relational Database Management System (RDBMS) is the format of most popular databases. Data is stored in tables and the relationships of the data is also stored in tables.

## OLAP



## Creating JNDI Data Connections

The instructions below explain how to add JNDI connections to your application server (which makes them available to the BA Server), and to the global JNDI configuration files for Pentaho design tools.



**Note:** JNDI data connections perform better and are more scalable and portable than JDBC or "Pentaho" data sources created through the Data Source Wizard. While Pentaho data sources are excellent for quick prototyping, you should switch to JNDI connections for production deployment, or if you need to share a data source across more than one deployed Web application.

### Adding a JNDI Data Connection to JBoss

If you must use JNDI connections instead of Pentaho-managed connections for your databases, follow the configuration instructions below. The preferred and much simpler method of database connection management is through the Pentaho Enterprise Console, but there may be instances where you want to share data connection settings with other Web applications.

1. Consult your database documentation to determine the class name and connection string for your database.
2. Stop the JBoss server.
3. Now you need to specify a database and set a JNDI name to reference it by. These properties are specified in a data source XML file that you will create specifically for your database. Navigate to the `/jboss/server/default/deploy/` directory.
4. Open a text editor and create a file named **my\_jndi\_datasource-ds.xml**, and use this example code as a basis for its content, changing the database type, address, port, and database name, the driver class for your database (if it isn't MySQL, as in the example), and the access credentials:

```
<?xml version="1.0" encoding="UTF-8"?>
<datasources>
 <local-tx-datasource>
 <!-- JNDI name of the datasource, it is prefixed with java:/ -->
 <jndi-name>my_jndi_datasource</jndi-name>
 <connection-url>jdbc:mysql://localhost:3306/myDatabaseName</connection-url>
 <driver-class>org.gjt.mm.mysql.Driver</driver-class>
 <user-name>username</user-name>
 <password>password</password>
 </local-tx-datasource>
</datasources>
```

5. Next, edit the `/jboss/server/default/deploy/pentaho.war/WEB-INF/web.xml` file and search for this comment:

```
<!-- insert additional resource-refs -->
```

6. Below the comment, create a new resource-ref that follows this example, changing the necessary details as you did previously (the res-type remains the same for all databases, so you do not need to modify that property):

```
<resource-ref>
 <description>my_jndi_datasource</description>
 <res-ref-name>jdbc/my_jndi_datasource</res-ref-name>
 <res-type>javax.sql.DataSource</res-type>
 <res-auth>Container</res-auth>
</resource-ref>
```

7. Now edit the `jboss-web.xml` file in the same directory, and search for this comment:

```
<!-- insert additional jboss resource-refs -->
```

8. Below the comment, create a new resource-ref that follows this example, again changing the necessary details to match your database:

```
<resource-ref>
 <res-ref-name>jdbc/my_jndi_datasource</res-ref-name>
 <res-type>javax.sql.DataSource</res-type>
 <jndi-name>java:/my_jndi_datasource</jndi-name>
```

```
</resource-ref>
```

9. Delete the `/jboss/server/default/work/` and `/jboss/server/default/tmp/` directories.

These directories contain cached files from deployed WARs and EARs, including the files you've just changed. Since the caching system does not immediately detect changes, you have to force it to rebuild by deleting the old files.

10. Start the JBoss server

You should now have access to your database through the BA Server. If you have not already prepared your data for reporting and analysis through Metadata Editor and Schema Workbench, you should do so now if you intend to use any elements of Pentaho Reporting and Analysis.

## Adding a JNDI Data Connection to Tomcat

If you would prefer to use a JNDI connection to your database instead of the standard Pentaho method, follow the instructions below:

1. Consult your database documentation to determine the class name and connection string for your database.
2. Stop the Tomcat server.
3. Edit the `/tomcat/webapps/pentaho/WEB-INF/web.xml` file.
4. At the end of the **<web-app>** element (in the same part of the file where you see **<!-- insert additional resource-refs -->**), add the XML snippet shown below:

```
<resource-ref>
 <description>myDataSource</description>
 <res-ref-name>jdbc/myDataSource</res-ref-name>
 <res-type>javax.sql.DataSource</res-type>
 <res-auth>Container</res-auth>
</resource-ref>
```

Change the **description** and **res-ref-name** nodes (and any others that apply to your situation; you may need to consult <http://tomcat.apache.org/tomcat-6.0-doc/jndi-datasource-examples-howto.html> to see if there are other things to consider) to fit your data source

5. Save and close the `web.xml` file.
6. Edit the `/tomcat/conf/context.xml` with a text editor.



**Note:** Alternatively, you can modify the `/tomcat/webapps/pentaho/META-INF/context.xml` file if you want this data source to be available only to the Pentaho BA Server. Adding JNDI connections to `context.xml` makes them available to all of the webapps deployed to this Tomcat instance.

7. Anywhere inside of the **<Context>** element, add the XML snippet shown below:

```
<Context path="/pentaho" docbase="webapps/pentaho/">
 <Resource name="jdbc/myDataSource" auth="Container" type="javax.sql.DataSource"
 factory="org.apache.commons.dbcp.BasicDataSourceFactory" maxActive="20"
 maxIdle="5"
 maxWait="10000" username="dbuser" password="password"
 driverClassName="org.postgresql.Driver"
 url="jdbc:postgresql://127.0.0.1:5432/myDataSource" />
</Context>
```

The above example shows a simple PostgreSQL configuration. Replace the **Resource name**, **username**, **password**, **driverClassName**, and **url** parameters (or any relevant connection settings) to match your database connection information, and the details you supplied in the `web.xml` file earlier.

8. Save and close the `context.xml` file.
9. Delete the `/tomcat/conf/catalina/` directory.

This directory contains a cached copy of the `web.xml` file you modified. The cache is not usually configured to update frequently, so you'll have to delete it and let Tomcat recreate it when it starts up.

10. Start the Tomcat server.

Tomcat can now properly connect to your database.

## Adding a Simple JNDI Data Source For Design Tools

Pentaho provides a method for defining a JNDI connection that exists only for locally-installed client tools. This is useful in scenarios where you will be publishing to a BA Server that has a JNDI data source; if you establish the same JNDI connection on your client tool workstation, you will not need to change any data source details after publishing to the BA Server. Follow the directions below to establish a simple JNDI connection for Pentaho client tools.

1. Navigate to the **.pentaho** directory in your home or user directory.

For a user name of **fbeuller**, typically in Linux and Solaris this would be `/home/fbeuller/.pentaho/`, and in Windows it would be `C:\Users\fbeuller\.pentaho\`

2. Switch to the `~/ .pentaho/simple-jndi/` subdirectory. If it does not exist, create it.
3. Edit the **default.properties** file found there. If it does not exist, create it now.
4. Add a data source by declaring a JNDI name followed by a forward slash, then a JNDI parameter and its proper value.

Refer to [Simple JNDI Options](#) on page 19 for more information on parameter options.

```
SampleData/type=javax.sql.DataSource
SampleData/driver=org.hsqldb.jdbcDriver
SampleData/user=pentaho_user
SampleData/password=password
SampleData/url=jdbc:hsqldb:mem:SampleData
```

5. Save and close the file.

You now have a global Pentaho data source that can be used across all of the client tools installed on this machine. You must restart any running Pentaho program in order for this change to take effect.

### Simple JNDI Options

Each line in the data source definition must begin with the JNDI name and a forward slash (/), followed by the required parameters listed below.

Parameter	Values
type	<b>javax.sql.DataSource</b> defines a JNDI data source type.
driver	This is the driver class name provided by your database vendor.
user	A user account that can connect to this database.
password	The password for the previously declared user.
url	The database connection string provided by your database vendor.

```
SampleData/type=javax.sql.DataSource
SampleData/driver=org.hsqldb.jdbcDriver
SampleData/user=pentaho_user
SampleData/password=password
SampleData/url=jdbc:hsqldb:mem:SampleData
```

## Creating JDBC Database Connections

The instructions in this section explain how to add JDBC connections to a Pentaho data sources through both the Pentaho Enterprise Console and the Pentaho User Console using the Pentaho Data Source Wizard. For a list of supported data sources and JDBC drivers, see the *Compatibility Matrix: Supported Components* in any of the install guides.



**Note:** JNDI data connections perform better and are more scalable and portable than JDBC data sources created through the Data Source Wizard. While Pentaho data sources are excellent for quick prototyping, you should switch to JNDI connections for production deployment.

## Adding a JDBC Driver

Before you can connect to a data source in any Pentaho server or client tool, you must first install the appropriate database driver. Your database administrator, Chief Intelligence Officer, or IT manager should be able to provide you with the proper driver JAR. If not, you can download a JDBC driver JAR file from your database vendor or driver developer's Web site. Once you have the JAR, follow the instructions below to copy it to the driver directories for all of the Business Analytics components that need to connect to this data source. See the *Compatibility Matrix: Supported Components* in any of the Installation guide for current version numbers.



**Note:** Microsoft SQL Server users frequently use an alternative, non-vendor-supported driver called JTDS. If you are adding an MSSQL data source, ensure that you are installing the correct driver.

### Backing up old drivers

You must also ensure that there are no other versions of the same vendor's JDBC driver installed in these directories. If there are, you may have to back them up and remove them to avoid confusion and potential class loading problems. This is of particular concern when you are installing a driver JAR for a data source that is the same database type as your Pentaho solution repository. If you have any doubts as to how to proceed, contact your Pentaho support representative for guidance.

### Installing JDBC drivers

Copy the driver JAR file to the following directories, depending on which servers and client tools you are using (Dashboard Designer, ad hoc reporting, and Analyzer are all part of the BA Server):



**Note: For the DI Server:** before copying a new JDBC driver, ensure that there is not a different version of the same JAR in the destination directory. If there is, you must remove the old JAR to avoid version conflicts.

- **BA Server:** /pentaho/server/biserver-ee/tomcat/lib/
- **Enterprise Console:** /pentaho/server/enterprise-console/jdbc/
- **Data Integration Server:** /pentaho/server/data-integration-server/tomcat/webapps/pentaho-di/WEB-INF/lib/
- **Data Integration client:** /pentaho/design-tools/data-integration/libext/JDBC/
- **Report Designer:** /pentaho/design-tools/report-designer/lib/jdbc/
- **Schema Workbench:** /pentaho/design-tools/schema-workbench/drivers/
- **Aggregation Designer:** /pentaho/design-tools/agg-designer/drivers/
- **Metadata Editor:** /pentaho/design-tools/metadata-editor/libext/JDBC/



**Note:** To establish a data source in the Pentaho Enterprise Console, you must install the driver in both the Enterprise Console and the BA Server or Data Integration Server. If you are just adding a data source through the Pentaho User Console, you do not need to install the driver to Enterprise Console.

### Restarting

Once the driver JAR is in place, you must restart the server or client tool that you added it to.

### Connecting to a Microsoft SQL Server using Integrated or Windows Authentication

The JDBC driver supports Type 2 integrated authentication on Windows operating systems through the **integratedSecurity** connection string property. To use integrated authentication, copy the **sqljdbc\_auth.dll** file to all the directories to which you copied the JDBC files.

The **sqljdbc\_auth.dll** files are installed in the following location:

```
<installation directory>\sqljdbc_<version>\<language>\auth\
```



**Note:** Use the **sqljdbc\_auth.dll** file, in the x86 folder, if you are running a 32-bit Java Virtual Machine (JVM) even if the operating system is version x64. Use the **sqljdbc\_auth.dll** file in the x64 folder, if you are running a

64-bit JVM on a x64 processor. Use the `sqljdbc_auth.dll` file in the IA64 folder, you are running a 64-bit JVM on an Itanium processor.

## Adding a JDBC Database Connection In Enterprise Console

You must have the appropriate JDBC driver installed in Enterprise Console and the BA Server in order to establish a data source in Enterprise Console.

To define a Pentaho database connection:

1. In the Pentaho Enterprise Console go to **Administration > Database Connections**.
2. Click the **General** icon to display basic configuration options.
3. Click the plus sign (+) (add) if you cannot find your database in the default list.  
The **Add Database Connection** dialog box appears.
4. Type a connection name that concisely describes the data source.
5. Type or select the **Driver Class** from the list. 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 HSQLDB database.



**Note:** If your JDBC driver is not available, see [Adding a JDBC Driver](#).

6. Type the **User Name** and **Password** required to access your database.
7. Type or select the **URL** from the list. This is the URL of your database; for example, `jdbc:hsqldb:hsq://localhost/sampledatab`. JDBC establishes a connection to a SQL-based database and sends and processes SQL statements.
8. Click **Test**.  
A success message appears if the connection is established.
9. Click **OK** to save your entries.

## Using Advanced JNDI Connection Options in the Enterprise Console

In the Pentaho Enterprise Console go to **Administration > Database Connections** and select the data source, then select the **Advanced** option.

In this section you can define several options:

Data Source Type	Description
Maximum Active Connections	The maximum number of active connections that can be allocated from this pool at the same time, or negative for no limit. <b>The default value is 20.</b>
Number of Idle Connections	The number of connections that can remain idle in the pool, without extra ones being released, or negative for no limit. <b>The default value is five.</b>
Validation Query	The SQL query that will be used to validate connections from this pool before returning them to the caller. If specified, this query <b>MUST</b> be an SQL <b>SELECT</b> statement that returns at least one row. <b>By default, this is blank.</b>
Wait: (milliseconds)	The number of milliseconds that the pool will wait (when there are no available connections) for a connection to be returned before throwing an exception, or -1 to wait indefinitely. <b>The default value is 100.</b>

If you edit values through the **Advanced** tab, you will only edit on an individual data source basis. Global defaults are stored in the BA Server `pentaho.xml` file, which can be found in `/pentaho-solutions/system/pentaho.xml`.

## Adding a Data Source in the Pentaho User Console

The Data Sources feature allows you to connect to your data so that the data can be used to create reports, (such as Interactive Reports, Analyzer Reports, and Dashboards), in the Pentaho User Console.

Using the Data Source Wizard, you can quickly add, edit, and delete **CSV File**, **SQL Query**, and **Database Table(s)** data sources in the Pentaho User Console.

Below is a quick description of each data source type:

Data Source Type	Description
CSV File	Data originating in a CSV file is extracted and staged in a database table on the Pentaho BA Server. A default Pentaho Metadata (Reporting) Model and a Mondrian (OLAP) Schema are generated for use in Interactive Report, Dashboard, and Analyzer.
SQL Query	A SQL query written against a relational database is used to establish the context of the data source. Data available for report creation is confined to the scope of the data source's query. A default Pentaho Metadata (Reporting) Model and a Mondrian (OLAP) Schema are generated for use in Interactive Report, Dashboard, and Analyzer.
Database Table(s) Reporting Only	Data originates in a one or more relational database tables that is often operational or transactional in nature. Using the Data Source Wizard, database tables can be selected and joined. A default Pentaho Metadata (Reporting) Model is generated for use in Interactive Report and Dashboard only.
Database Table(s) Reporting and Analysis	Data originates in one or more relational database tables arranged in a star schema with a single fact table. Using the Data Source Wizard, multiple dimension tables can be selected and joined to the single fact table. A single table containing both fact and dimensional information can also be used for this data source type. A default Pentaho Metadata (Reporting) Model and a Mondrian (OLAP) Schema are generated for use in Interactive Report, Dashboard, and Analyzer.

You can restrict access to a data source by individual user or user role; by default, authenticated users are limited to view-only permissions. (See "Assigning Data Source Permissions for the Pentaho User Console" in the **Pentaho Security Guide** for more information.) This means that they are limited to selecting a data source from a list of available options.



**Note:** Non-administrative users of the Pentaho User Console do not see icons or buttons associated with adding, editing, or deleting a data source.

The **Data Source Wizard**, which allows you to set up access to a relational data source or CSV data source in the Pentaho User Console, is available in the following instances:

- When you click the **New Data Source** button on the Pentaho User Console Quick Launch page
- When you select **File -> New -> Data Source** in the Pentaho User Console menu bar
- When you select **File -> Manage -> Data Sources** from the Pentaho User Console menu bar and click the Add Icon
- When you click the Add Icon in the **Select Data Source** dialog box when creating an Interactive Report.
- When you click the Add Icon in the **Select Data Source** dialog box when creating a chart in a Dashboard
- When you click the Add Icon in the **Select Data Source** dialog box when creating a data table in a Dashboard
- When you click **Create New Data Source** when creating a new Analyzer report



**Note:** Data Source Wizard is intended for prototyping or as a starting point for Pentaho Analysis and Pentaho Metadata models used with Interactive Reporting reports, Analyzer reports, and dashboards. We recommend that these models be re-created or edited with Schema Workbench or Metadata Editor to ensure query performance and to take advantage of enhanced functionality when used in production environments.




**Note:** The following characters are not allowed in Data Source Wizard source names: \$ < > ? & ; # % ^ \* ( ) ! ~ : ; [ ] { } |



## Creating a SQL Query in the Pentaho User Console

Follow the instructions below to create a SQL Query data source. In addition to allowing you to connect to your database (using JDBC), you must also enter a SQL statement that defines the "scope," (similar to a database view), of your data source.


 **Note:** In the context of the Pentaho User console, a SQL Query "data source" is a combination of a connection plus a query. Pentaho recommends that you use the Pentaho Metadata Editor to create SQL Query data sources for a production environment.

You can customize how columns are presented to users who are building queries against the new data source; for example, you can define "friendly" column names and select options that define how data is aggregated (sum, min., max., etc.), and more.

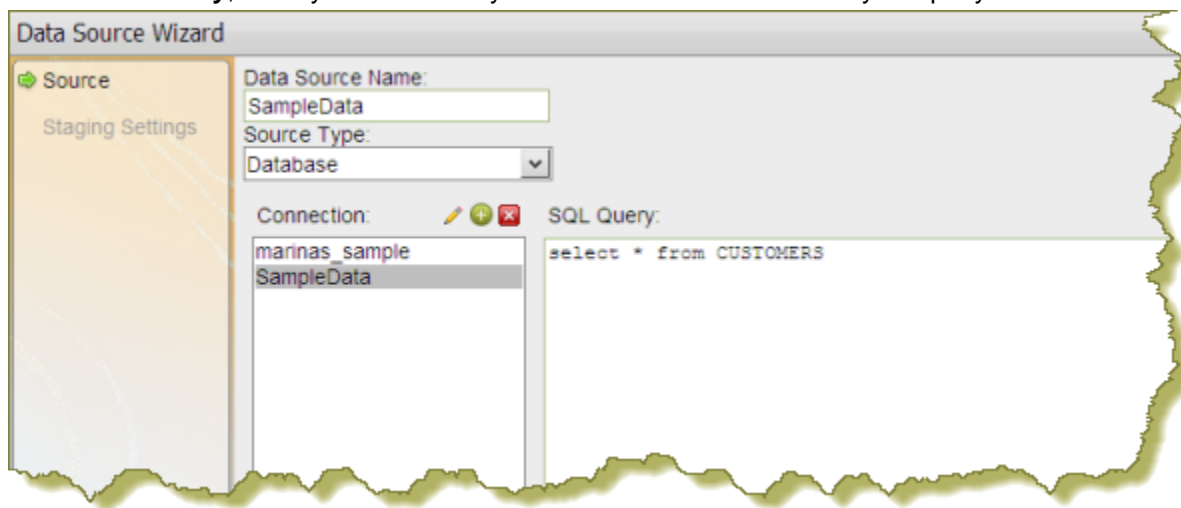
Once you create the data source, it is available to users of the Interactive Reports, Analyzer, and Dashboards.

You must be logged on to the Pentaho User Console as an administrator to follow the instructions below. Additionally, your database must be up and running in order for you to complete the steps required in the Data Source Wizard.

1. In the **Data Source Wizard**, type a name for the new data source in the **Data Source Name** field.

 **Note:** The name of the data source is important because it becomes the name of the data source displayed in Analyzer, Interactive Report, and Dashboards.

2. Under **Source Type**, select **SQL Query** as your data source type.
3. Select a database connection from the list. under **Connection**.
4. Under **SQL Query**, enter your SQL Query. Click **Preview** to ensure that your query returns data.



5. Click **Close** to exit the Preview window.


6. Click **Finish**.

In the background, after you click **Finish**, a default metadata model is generated based on the query for use in Analyzer, Interactive Report, and Dashboards. See [Editing a Data Source Connection](#) for instructions about changing connection details.

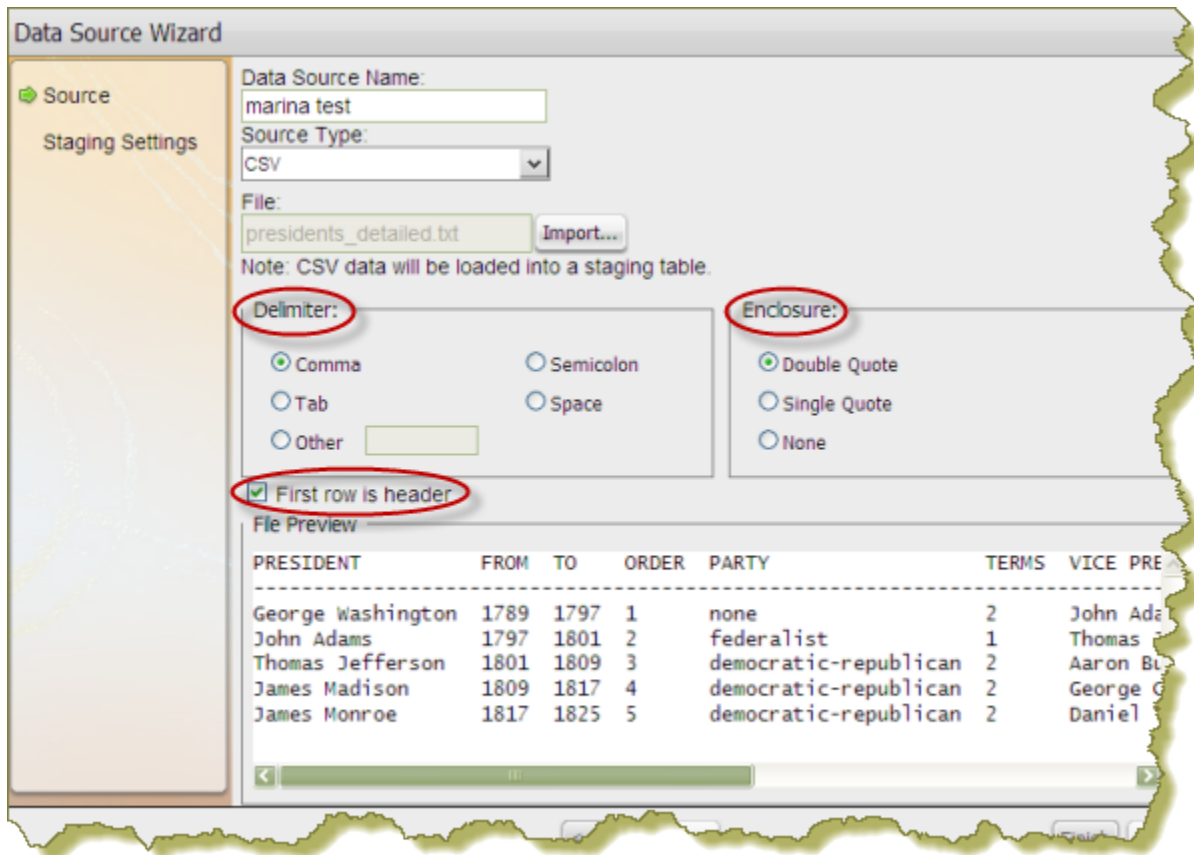
## Creating a CSV Data Source in the Pentaho User Console

Follow the instructions below to create a CSV-based data source. For testing purposes, you can save an Excel spreadsheet as a CSV file.

1. In the **Data Source Wizard**, type a name that identifies your new data source in the **Data Source Name** field.
2. Under **Source Type** select **CSV File** from the list to define a CSV file as your data source and click **Next**.
3. Click **Import** to open the **Import File** dialog box.
4. Click **Browse** to search for your CSV file; click **Import** when you locate the file.  
The Import File dialog box closes and the CSV file is uploaded to the BA Server. You can click **Cancel**, if necessary, and select different file.

 **Note:** Notice that a preview of the contents of your file appears in the lower portion of the wizard page. If items are not aligning correctly, it is because you must specify your delimiter and enclosure types. Go on to the next step.

- Specify the delimiter and enclosure types. Enable **First row is header** if the first row of your CSV file is a heading for columns in the file.



**Data Source Wizard**

Source

Staging Settings

Data Source Name: marina test

Source Type: CSV

File: presidents\_detailed.txt

Note: CSV data will be loaded into a staging table.

Delimiter:

☒ Comma ☐ Semicolon

☐ Tab ☐ Space

☐ Other

Enclosure:

☒ Double Quote


☐ Single Quote

☐ None

☒ First row is header


File Preview

PRESIDENT	FROM	TO	ORDER	PARTY	TERMS	VICE PRES
George Washington	1789	1797	1	none	2	John Adams
John Adams	1797	1801	2	federalist	1	Thomas Jefferson
Thomas Jefferson	1801	1809	3	democratic-republican	2	Aaron Burr
James Madison	1809	1817	4	democratic-republican	2	George Clinton
James Monroe	1817	1825	5	democratic-republican	2	Daniel Denny

 **Important:** You must make selections under **Delimiter** and **Enclosure** for the Metadata table to build correctly. In the example line, 1,4,5,02/12/77,"String Value" the commas are delimiters and the quotation marks around "String Value" is the enclosure.

The **File Preview** window displays the first few lines of your CSV file based on the selections you made for specifying the delimiter, enclosure, and header. Once the columns align correctly in the preview you will know that the delimiter and enclosure have been set correctly.

- Click **Next** to display more details about the columns in your source file and to edit staging-related settings.

 **Note:** Notice that all columns are enabled in this step. Click **Deselect All** to disable all columns. You can then select the columns that require formatting individually. Click **Select All** to enable all columns again, if necessary.

- If applicable, change the **Name** and **Type** values. Use the drop-down list to display format options for dates and numeric values. Formats are auto-detected based on the contents of your data source; however, if you see a format option that is not correct, enter it manually in the appropriate **Source Format** text box.

**Length** indicates the maximum number of characters allowed in a field. **Precision** is the number of digits after a decimal point. Drop-down lists are not enabled for certain data types such as the String data type.



**Data Source Wizard**

Source  
Staging Settings

Select the columns you want to stage on the Pentaho BI Server and configure their attributes. Use Source Format to specify how numeric or date fields are formatted in the source CSV file.

	Name	Type	Source Format	Length	Precision
<input checked="" type="checkbox"/>	ORDERNUMBER	STRING		13	0
<input checked="" type="checkbox"/>	QUANTITYORDERED	NUMERIC	#	4	0
<input checked="" type="checkbox"/>	PRICEEACH	NUMERIC	##	7	2
<input type="checkbox"/>	ORDERLINENUMBER	NUMERIC	### 00	4	0
<input checked="" type="checkbox"/>	SALES	STRING	###0	13	0
<input checked="" type="checkbox"/>	ORDERDATE	STRING	### #	22	0
<input checked="" type="checkbox"/>	STATUS	STRING		15	0
<input checked="" type="checkbox"/>	QTR_ID	NUMERIC		3	0
<input checked="" type="checkbox"/>	MONTH_ID	NUMERIC		4	0
<input checked="" type="checkbox"/>	YEAR_ID	STRING		12	0
<input checked="" type="checkbox"/>	PRODUCTLINE	STRING		18	0

Show File Contents



**Important:** Boolean values are rendered as "true" or "false."

- Click **Show File Contents** to see a sample of the data in your source file. Click **OK** to return to staging and make changes if necessary.
- Click **Finish** when you have completed your configuration and formatting tasks.  
A success message appears indicating the number of rows being loaded into the database and that a default metadata model was created. Enable **Keep default model** to use the default model or click **Customize model now** to customize the model.

The data in your CSV file is extracted and staged in a database table. Your new data source is now available for use in Analyzer, Interactive Report, Ad Hoc Report, and Dashboards. To refine the data and make it more accessible to users of you must use the Data Source Modeler. The model can be further customized to adjust field names, formatting, aggregation types, and dimensional hierarchies (for Pentaho Analysis). See [Editing a Data Source in the Pentaho User Console](#).

### Increasing the CSV File Upload Limit

Follow the instructions below if you need to increase the size of the upload limit associated with CSV files.

- Go to ...\\biserver-ee\\pentaho-solutions\\system and open the **pentaho.xml** file.
- Edit the XML as needed; (sizes are measured in bytes):

```
<file-upload-defaults>
 <relative-path>/system/metadata/csvfiles/</relative-path>

 <!-- max-file-limit is the maximum file size, in bytes, to allow to be
 uploaded to the server -->
 <max-file-limit>10000000</max-file-limit>

 <!-- max-folder-limit is the maximum combined size of all files in the upload
 folder, in bytes. -->
 <max-folder-limit>500000000</max-folder-limit>

</file-upload-defaults>
```

- Save your changes to the file.

4. In the **Pentaho User Console**, go to **Tools -> Refresh System Settings** to ensure that the change is implemented.
5. Restart the Pentaho User Console.

### Changing the Staging Database

Hibernate is the default staging database for CSV files. Follow the instructions below if you want to change the staging database.

1. Go to ... \pentaho-solutions\system\data-access and open **settings.xml**.
2. Edit the settings.xml file as needed. The default value is shown in the sample below.

```
<!-- settings for Agile Data Access -->
<data-access-staging-jndi>Hibernate</data-access-staging-jndi>
```



**Note:** This value can be a JNDI name or the name of a [Pentaho Database Connection](#).

3. Save your changes to the **settings.xml** file.
4. Restart the Pentaho User Console.

### Where Metadata Models and Mondrian Schemas are Stored

All generated Metadata Models and Mondrian Schemas are stored in the following location, \pentaho-solutions\admin\resources\metadata.



**Important:** Pentaho recommends that you *do not* change the location where models and schemas are stored.

### Customizing an Analysis Data Source

The Modeler allows you to further refine your analysis-based data source. For example, you can change the field display names, create hierarchies and levels, set an aggregation type, and change formatting.



**Note:** If you have created a CSV, SQL Query, or a Database Table(s) Reporting and Analysis data source, you can make changes to both the Reporting metadata model and to the Analysis model (Mondrian Schema), independently.




**Note:** While users are able to create interactive reports, Analyzer reports, and dashboards using the default model that is generated when you first create a data source, the structure of the default Analysis model is "flat." For example, the auto-model process creates a dimension for every table and a hierarchy plus level combination for all columns in that table. In most instances you will want to refine the default model to make the data more easily understandable to your users.

Follow the instructions below to edit an analysis data source:


1. In the **Data Source Modeler**, examine your data.

Measures are numeric values such as sales price, cost, and profit. **Dimensions** are the context that allow users to understand the significance of the measures. For a sales database, the dimensions could include Product, Store, Salesperson and so on. In other words, users can examine the sales figures by store, by product, and by sales.

Dimensions can have one or more hierarchies (). Hierarchies contain levels that organize data into a logical structure. Click (+) next to fields under Measures and Hierarchies to display their contents. Moving up and down between the levels of a hierarchy is called *drilling*.






**Note:** Measures must originate in the fact table; for example, you cannot move numeric fields in a dimension table under **Measures** or an error message is displayed. You cannot place two levels that originate in different tables into a single hierarchy.

Unnecessary fields can be removed by clicking  (Delete) in the modeler toolbar.

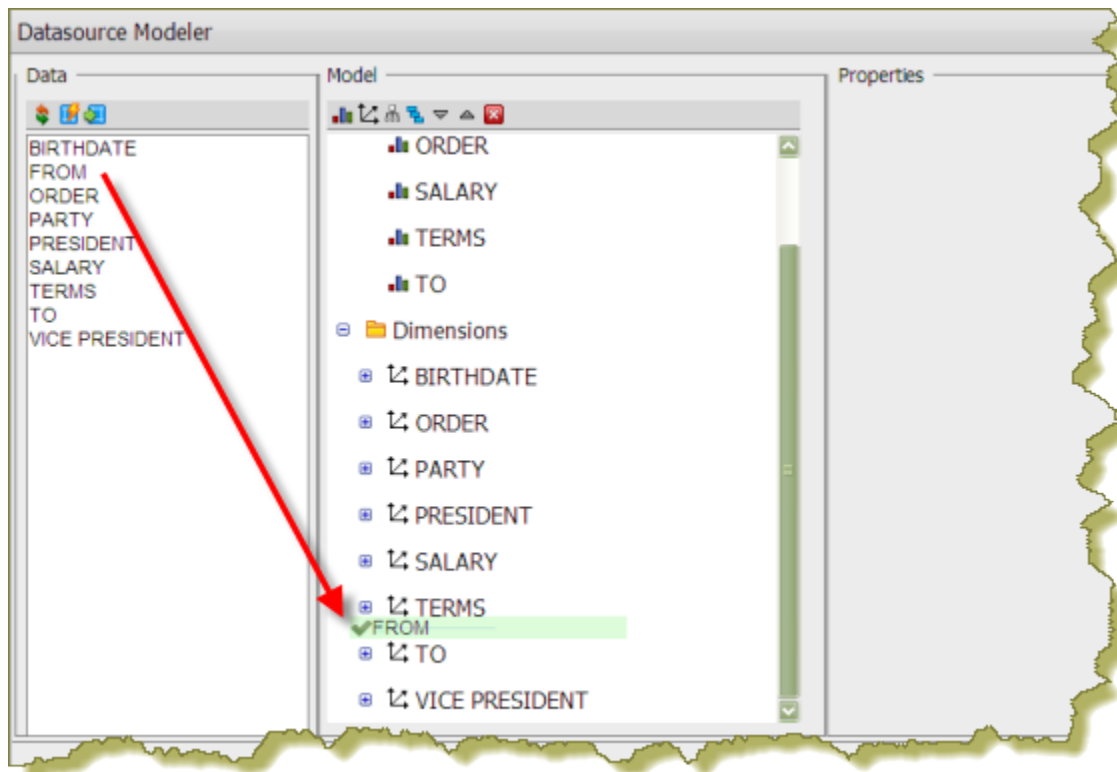
You can move a field under Measures or Dimensions by clicking and dragging the field into its appropriate location. A green check mark indicates that you can place your field without error.



**Note:** Click  (Clear Model, in the upper right corner) to clear the existing model and start over.

 **Note:** If necessary, click  (Edit Source) in the modeler to make changes to database table selections and joins, edit a SQL query, or CSV flat file. Click **Finish** to save your updates. If you disable columns that were previously enabled, a prompt appears in the modeler indicating that the data has been deleted. You must adjust the data in the modeler before proceeding.

You can also move a field under Measures or Dimensions by clicking and dragging the field into its appropriate location.




2. Click **OK** to update the data source.

The updated data source is now available to users.


### Editing a Database Connection in the Pentaho User Console


Follow the instructions below to edit a database connection in the Pentaho User Console:

1. In the **New Data Source** dialog box, under **Connection**, select the connection you want to edit.
2. Click .  
The Database Connection dialog box appears.
3. Update the connection information as needed. You can click **Test** to make sure your connection is working correctly.
4. Click **OK** to exit the Database Connection dialog box.

### Deleting a Database Connection in the Pentaho User Console


Follow the instructions below to delete a database connection associated with the Pentaho User Console.

 **Important:** If you delete a database connection, all reports, charts, and other content users have created using the connection will no longer render.




1. Under **Connection** in the **New Data Source** dialog box, select the connection you want to delete.
2. Click  (Delete).  
A warning dialog box appears.
3. Click **OK** to delete the connection.

## Deleting a Data Source in the Pentaho User Console

If you have administrative permissions, you can delete data sources associated with ad hoc reports and dashboards in the Pentaho User Console.

 **Important:** When you delete a data source all reports or charts that end-users have created based on that data source will no longer render. Deleting a data source also deletes the model associated with that data source.

The delete Data Source option is available in the following instances:

- When you select a data source and click  (Delete) when in the **Select Data Source** dialog box available from **File -> Manage -> Data Sources**
- When you select a data source and click **Delete** in the **Select Data Source** step in Ad Hoc Reports
- When you select a data source and click  (Delete) when creating a chart
- When you select a data source and click  (Delete) when creating a data table

## Adjusting Name Restrictions in Data Source Wizard


The Data Source Wizard's default configuration prohibits the following characters from being used in a data source name:

- \$
- <
- >
- ?
- &
- #
- %
- ^
- \*
- (
- )
- !
- ~
- :
- ;
- [
- ]
- {
- }
- |

You can add to these restrictions by editing **<data-access-datasource-illegal-characters>** node in the `/pentaho/server/biserver-ee/pentaho-solutions/system/data-access/settings.xml` file.

## Restricting Metadata Models to Specific Client Tools

To prevent a published metadata model from being displayed in the data source lists of one or more Pentaho User Console client tools (Dashboard Designer, Interactive Reporting, and ad hoc reporting), follow this process:

 **Note:** ROLAP data sources for Analyzer and JPivot are not affected by this configuration.

1. Start Metadata Editor and open the metadata model you want to restrict.
2. Right-click on the business model, then select **Edit** from the context menu.  
The **Business Model Properties** window will appear.
3. Click the round green **+** icon to add a new custom property.  
The **Add New Property** window will appear.

4. Click **Add a custom property**, then type **visible** in the **ID** field and select **String** from the **Type** drop-down box, then click **OK**.

You'll return to the **Business Model Properties** window.

5. Scroll down to the bottom of the window to see the **Value** field. Type in one or more of the following restriction values, separated by commas, then click **OK**:

- **adhoc**: removes the model from the list in ad hoc reporting
- **dashboard-filters**: prevents this model from appearing in dashboard filters that use metadata queries
- **dashboard-content**: prevents this model from being used in dashboard panels
- **interactive-reporting**: removes the model from the list in Interactive Reporting
- **manage**: removes this model from the "manage data sources" feature of the Pentaho User Console



**Note:** If you leave the **Value** field blank, the published model will not appear in any Pentaho User Console tools.

6. Re-publish this model to the BA Server.
7. If the BA Server is currently running, restart it.

The restrictions you defined should now be in place.

## Managing Schedules

---

The Scheduler allows you to create, update, delete, run, suspend, and resume one or more schedules, (Public and Private), in the BA Platform. In addition, you can suspend and resume the Scheduler itself. In the context of the BA 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 sales report to which a manager or salesperson can subscribe. As the administrator, the schedule (or schedules) you designate determines when the Scheduler allows the action sequence to run. Private schedules are ad hoc, non-subscription schedules, which are associated with one action sequence only.



**Note:** Public Schedules were formerly called, "subscription schedules;" private schedules were formerly called, "regular schedules."

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.

Why not allow BA Platform users to create schedules whenever they want? Allowing that much flexibility may, among other things, overload servers. In most instances, you know when it is most sensible to schedule an action sequence to run; for example, after all stores upload their sales figures. In other instances, sales data may not change for a week or month, so reporting hourly would not make sense. You, or a solution developer, can define as many schedules as needed for a specific action sequence. Users are allowed to choose from schedules that make sense to them and you can schedule the run to occur at a time of minimal load.

## Entering Schedules in the Schedule Creator Dialog Box

---

Enter schedules associated with your action sequences in the Schedule Creator dialog box. The Schedule Creator makes it easy for you to enter schedules without having to learn the syntax of Cron expressions; however, it provides you with the option to enter Cron expressions if that is your preference.

The scheduling process is not complete until you associate the schedule with one or more action sequences. If you are creating a private schedule, you are limited to associating the schedule to a single action sequence.

To enter schedules...

1. In the main page of the Pentaho Enterprise Console, click **Administration**.
2. Click the **Scheduler** tab.
3. In the Scheduler, click the New icon to open the **Scheduler Creator** dialog box.



**Note:** By default, the Scheduler creates a public schedule. To create a private schedule, disable the **Make Public** check box.

4. Under Schedule, enter a **Name** for the schedule, for example, Monthly Sales.
5. Enter a **Group** associated with the schedule, for example, Sales Schedules.
6. Enter a short **Description** of the schedule. for example, "Schedule runs on the first of each month, schedule runs on Monday of each week."
7. Select a **Recurrence Type**. You can schedule the action sequence to run once at a particular date and time only, or have it recur in seconds, minutes, hours, daily, weekly, monthly, yearly, or recur based on a Cron string. The options in the Recurrence Editor change depending on the type of recurrence you select.



**Note:** You can use the Schedule Creator to enter a Cron expression manually by selecting Cron from the Recurrence Type list. See *CRON Expressions in Detail* (link below) to learn more about Cron expressions.

You must now associate the new schedule with one or more action sequences.

## Associating Schedules with Action Sequences

---

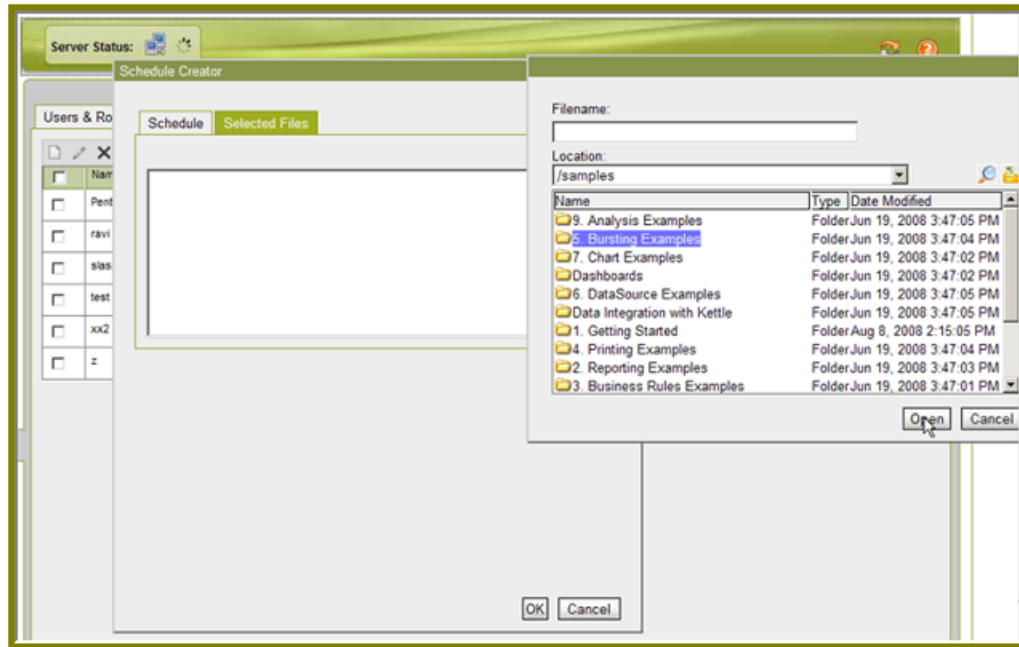
After you add your schedules, you must associate them with action sequences.



**Caution:** For private schedules, you must associate a schedule with a minimum of one action sequence or an error message appears.

To associate schedules with action sequences...

1. In the **Schedule Creator** dialog box, click the **Selected Files** tab.
2. Click the Add button next to **Associate this public schedule with the selected files** to open the **Select** (file chooser) dialog box.
3. In the **Select** dialog box double-click a folder to display contents or select the folder and click **Open**.



4. Choose the action sequence file you want and click **Select**.  
The action sequence file appears under **Selected Files**. You can add more action sequence files as needed by repeating the process above.







## Removing Action Sequence Files from a List

A schedule can be associated multiple action sequence files. To remove an action sequence file from a list, select the file and click the minus icon. To remove multiple action sequence files, press **CTRL + CLICK** to select the files for deletion.

## Examining the List of Schedules

As you create new schedules, the schedules appear in a list box. By examining the list, you can identify the **Name** and **Group** associated with each schedule. You can also determine the status (**State**) of each schedule and read a brief description of the schedule. In addition, you can determine when the schedule was first run (**Fire Time - Last/Next**) and when it will run again. The controls on the top corners of the Scheduler page define the tasks you can perform using the Scheduler:

Icon	Control Name	Function
	<b>Create Schedule</b>	Allows you to create a new schedule
	<b>Edit Schedule</b>	Allows you to edit the details of a schedule
	<b>Delete Schedule</b>	Allows you to delete a specified schedule; however, if the schedule is currently executing in a scheduler thread it continues to execute but no new instances are run

Icon	Control Name	Function
		 <b>Note:</b> To reliably delete a public schedule, (in the Pentaho User Console), users must suspend the schedule first then delete it from their workspace. (Users must have administrative privileges to delete a schedule.) After the delete process completes, they may restart the schedule.
	<b>Suspend Schedule</b>	Allows you to pause a specified schedule; once the job is paused the only way to start it again is with a Resume Schedule
	<b>Resume selected Schedule(s)</b>	Allows you to resume a previously suspended schedule; once the schedule is resumed the Scheduler applies misfire rules if needed
	<b>Run selected Schedule(s) Now</b>	Allows you to run a selected schedule(s) at will
	<b>Refresh</b>	Allows you to refresh the list of schedules
Filter By: 	<b>Filter by</b>	Allows you to search for a specific schedule by group name



# Configuring the BA Server

---

This section contains instructions for configuring and testing your BA server settings.

## Software Updates

---

To configure your software updates...

1. In the Pentaho Enterprise Console home page, click **Configuration**.
2. Click the **General** tab.
3. In the **Global Settings** group box, update the version-related settings as shown below:

Item	Description
<b>Version Check Interval</b>	Interval, in seconds, to check for software updates
<b>Version Check Flags</b>	Flags representing the release types in which you are interested such as major or minor version updates
<b>Disable Version Check</b>	Disables the version check

4. Click **Submit** when you have completed your changes to these settings.  
A success message appears.

## Email Settings

---

To configure email settings:

1. In the Pentaho Enterprise Console home page, click **Configuration**.
2. Click the **BA Components** tab
3. Enter in your settings as described below:

Setting	Description
<b>Use Authentication</b>	Enable if the email server requires the sender to authenticate.
<b>Use Start TLS</b>	Disabled in most instances except when using Gmail. Enables the use of the STARTTLS command (if supported by the server) to switch the connection to a TLS-protected connection before issuing any logon commands. Note that an appropriate trust store must be configured so that the client will trust the server's certificate. Defaults to disabled.
<b>Default from address</b>	The address from which emails sent by the BA Server originate.
<b>User ID</b>	ID used to connect to the email server for sending email.
<b>SMTP Host</b>	Address of your SMTP email server for sending email.
<b>SMTP Port</b>	Port of your SMTP email server; Usually this value is 25, for Gmail the value is 587.
<b>SMTP Protocol</b>	Transport for accessing the email server. Usually this is SMTP. For Gmail this is SMTP.
<b>Use SSL</b>	Enable if the email server requires an SSL connection. For Gmail this value must be enabled.
<b>Enable Debugging</b>	Enable to output email-related debugging information from the JavaMail API to the BA server log file.

4. Click **Submit**.  
A success message appears.

## Specifying Your Publisher Password

The publisher password enables client tools such as the Pentaho Report Designer and the Pentaho Metadata Editor to publish content on the BA Server.

To specify your publisher password...

1. In the Pentaho Enterprise Console home page, click **Configuration**, then click the **General** tab.
2. In the **Publisher** group box, enter the old password in the **Old Password** text box; enter the new password in the **New Password** text box.
3. Click **Submit** to update the password.  
A success message appears.

## Specifying Global and Session Variables

To specify global and session variables...

1. In the Pentaho Enterprise Console home page, click **Configuration**.
2. Click the **General** tab.
3. In the **System Actions** group box, there are global and session action sequences that run once at Pentaho startup and once per user session, respectively. Use the controls to add, delete, and edit existing global and session action sequences. The table below provides description associated with action sequences:

Item	Description
<b>Action Sequence Path</b>	Solution path to the action sequence
<b>Scope</b>	Either Global or Session
<b>Type</b>	Either Servlet or Portlet

## Changing the Location of the Server Log File

If you used the install wizard to install Pentaho Business Analytics, the pentaho.log is written to the C:\WINDOWS\system32 directory. To change the location of the pentaho.log file, you must edit the log4j.xml available at /pentaho/server/biserver-ee/tomcat/webapps/pentaho/WEB-INF/classes/.

Modify the location as shown in the sample below using the appropriate path to your installation of Pentaho Business Analytics.

```
<param name="File" value="C:\Program Files\pentaho\server\biserver-ee\logs
\pentaho.log"/>
<param name="Append" value="true"/>
```

## Changing the Tomcat Port

Follow the instructions below to change the port on which Tomcat is running:

1. Load ... \biserver-ee\tomcat\conf\server.xml.
2. Locate the following line:

```
<Connector URIEncoding="UTF-8" port="8080" ...
```

3. Change the port value as needed.
4. Save the **server.xml** file and restart Tomcat.

After changing the port number for Tomcat, you must modify the BA Server **fully-qualified-server-url** to use the new port number. You can do this either in the Pentaho Enterprise Console or directly in the BA Server web.xml file.

In the ...pentaho\WEB-INF\web.xml, change the port number here:

```
<context-param>
 <param-name>fully-qualified-server-url</param-name>
 <param-value>http://localhost:8080/pentaho/</param-value>
</context-param>
```

In the Pentaho Enterprise Console, go to **Configuration -> Web Settings** and type the correct path in the **Base URL** text field.

## Hiding Report Parameters In URLs

When displaying a parameterized report, the BA Server will show a parameter name and value in the report's URL. If you would like to hide report URL parameters from Pentaho User Console users, follow the instructions below.

1. Edit the /pentaho/server/biserver-ee/pentaho-solutions/system/custom/xsl/DefaultParameterForm.xsl file with a text editor.
2. In the following line, change the value of **select** from false to **true**:

```
<xsl:variable name="USEPOSTFORFORMS" select="'false'" />
```

3. Save and close the file.
4. Log into the Pentaho User Console, then go to the **Refresh** section of the **Tools** menu and click on **Refresh System Settings**.

Parameters will no longer be shown in a report's URL.

## Setting the Geo Map Rendering Wait Time

There is a static length of time that the BA Server waits for a Geo Map to finish rendering before the image is captured; this allows for all map tiles to be downloaded and points to be plotted. The default wait time is **1200 milliseconds**, and it is set through the **<map-export-javascript-delay>** node in the /pentaho-solutions/system/pentaho-geo/settings.xml file:

```
<map-export-javascript-delay>1200</map-export-javascript-delay>
```

## Adding Geo Map Support to Data Source Wizard



**Note:** The documentation in this section applies to both Data Source Wizard in the Pentaho User Console, and the modeler built into PDI.

### Automatically building geography dimensions

To support Geo Map visualizations in Analyzer, you must associate location data fields with a particular geographic role. Once you identify the fields that represent Continent, Country, State, City, Postal Code, etc., you can use the Geo Service plugin in the BA Server to look up the geographic information for that data point and plot it on a map.

Before the auto-modeler can work properly for this purpose, you must configure it to recognize geographic fields in the source tables and create a geography dimension for them. This configuration is done through the /pentaho-solutions/system/data-access/settings.xml file. The default configuration looks like this:

```
<!-- Geography settings -->
<geo>
 <roles>continent, country, state, city, postal_code</roles>
 <dimension-name>Geography</dimension-name>

 <continent>
 <aliases>continent</aliases>
 </continent>
```

```

<country>
 <aliases>country, ctry</aliases>
</country>

<state>
 <aliases>state, province, st, stateprovince</aliases>
 <required-parents>country</required-parents>
</state>

<city>
 <aliases>city, town</aliases>
 <required-parents>country,state</required-parents>
</city>

<postal_code>
 <aliases>zip, postal code, zip code</aliases>
 <required-parents>country</required-parents>
</postal_code>

<latitude>
 <aliases>lat, latitude</aliases>
</latitude>

<longitude>
 <aliases>long, lng, longitude</aliases>
</longitude>
</geo>

```

The possible nodes are:

- **<roles>**: a comma-separated list of the detectable geographic roles. Any field identified as **geo-role** during auto-modeling will be looked up in the Geo Service for centroid and/or polygon information when added to a Geo Map visualization.
- **<dimension-name>**: the default name of the geography dimension in the model. If more than one table is identified as having geo-fields, then multiple geo dimensions will be created (one for each table). In this case, the dimension names will be in the form of: table\_name\_dimension-name.
- Role tags such as: **<continent>**, **<country>**, **<state>**, **<city>**, **<postal\_code>**. For each distinct value in the **<roles>** tag, a corresponding tag is required. As shown in the example, the **<roles>** tag defines continent, country, state, city, postal\_code.
- **<aliases>**: a comma-separated list of names that match source data columns. From the example given above, the aliases defined for the state tag are: state, province, st, stateprovince. Any column in the source tables of a data source that exactly matches one these aliases will be marked as the geo-role state and added as part of the geography dimension/hierarchy when auto-modeling. This is required for each role tag.
- **<required-parents>**: a comma-separated list of geo-roles that are required to uniquely identify the given role. To uniquely identify a city, you need to also know the country and state in which it resides. This is not required.
- **<latitude>** and **<longitude>**: these are similar to the role tags because they also have the **<aliases>** child element. However, latitude and longitude are not auto-modeled with the others. If latitude and longitude fields are detected when auto-modeling, they are added as Mondrian schema member properties of the most likely field. If the field name ends with **\_latitude** or any of the aliases defined in the settings, then the first part of that field name (before the underscore) is used to try to match to another field available in the source table. So if you have fields named **Customer\_Latitude** and **Customer\_Longitude**, then this coordinate pair will try to be associated with a field named **Customer**. If no Customer field is found, then the field immediately preceding the latitude and longitude fields will get the association. When the latitude and longitude fields are first in the list of fields, then the field immediately following them will get this association. Only one latitude/longitude pair is supported in any single table.

## Custom geo-roles

If you want to add your own geo roles, there are a two important details that you must be aware of:

1. The left-to-right order of the fields is respected in the dimension hierarchy.
2. To add a new geo role, add your new role to the list, add a corresponding element tag, and define its aliases and required parents if it has any. This will build the geo dimension that you expect. However, unless you supply the service with the appropriate data, the new role won't show up on a map visualization.

## Dashboard Printing in a Headless Linux Environment

To enable dashboard printing for headless Linux operating systems, you must implement an Xvnc server. Implementing an Xvnc server requires additional installation steps.

For step-by-step instructions, see [Enabling Dashboard Printing on Headless CentOS and Red Hat](#) or [Enabling Dashboard Printing on Headless Ubuntu](#).

### Enabling Dashboard Printing on Headless CentOS and Red Hat

These instructions apply to both CentOS and Red Hat, and use CentOS for exemplary purposes.

To implement headless dashboard printing, you need access to the SWT library that is shipped with the dashboards plugin. The WebKit library depends on this library. The relevant packages on CentOS are `webkitgtk-devel-1.2.6-2.el6_0` and `webkitgtk-1.2.6-2.el6_0`.

1. Install the relevant CentOS packages by entering `$ yum install webkitgtk-devel-1.2.6-2.el6_0 webkitgtk-1.2.6-2.el6_0`.
2. If the VNC server is not installed, run `$ yum install tigervnc-server`.
3. To install a desktop environment, run `$ yum groupinstall Desktop`. If you need more information, see the CentOS website and search for *virtual network computing (VNC)*.
4. Start the VNC server with this command: `$ vncserver :1 -geometry 1600x1200 -localhost $ export DISPLAY=localhost:1`.
5. Start the BA Server with this command: `$ ./start-pentaho.sh`. The VNC server and the BA Server must be started by the same user.

Set the `-geometry` option to the width and height of the largest supported client resolution.

You can add the commands to run and configure the VNC server to the Pentaho startup script, or you can start the VNC server as a regular service, whichever you prefer.

### Enabling Dashboard Printing on Headless Ubuntu

These instructions show how to enable headless dashboard printing on Ubuntu Linux.

1. Install the relevant Ubuntu packages.

```
$ sudo apt-get install libwebkit-dev
$ sudo apt-get install libwebkit-1.0-2
$ sudo apt-get install libswt-gtk-3.5-jni
```

2. If the VNC server is not installed, run `$ sudo apt-get install vnc4server` to install it.
3. To install a desktop environment, run `$ sudo apt-get install ubuntu-desktop`.
4. Start the VNC server with this command: `$ vncserver :1 -geometry 1600x1200 -localhost $ export DISPLAY=localhost:1`.
5. Start the BA Server with this command: `$ ./start-pentaho.sh`. The VNC server and the BA Server must be started by the same user.

Set the `-geometry` option to the width and height of the largest supported client resolution.

You can add the commands to run and configure the VNC server to the Pentaho startup script, or you can start the VNC server as a regular service, whichever you prefer.

# Solution Repository and Cache Management

---

The Pentaho solution repository stores solution content, runtime content (including audit log data), and schedule information for your BA Server. It is critical that this repository be available and working correctly for your BA Server to function properly.

## Backing Up the Solution Repository

---

Enterprise Console has built-in functions for backing up and restoring the BA Server solution repository. Follow the instructions below to back up or restore all BA Server content, including ACLs.



**Note:** ACLs are hard-coded to **pentaho.xml** when you execute a backup operation from Enterprise Console. This means that after you perform a backup, every time you use the **Reset Permissions** function in the future, the default ACLs set by the most recent backup will be the new defaults unless you erase the hard-coded ACLs from pentaho.xml by hand.

1. Log into Enterprise Console.
2. Click the **Setup** button in the upper right corner of Enterprise Console.  
The **Pentaho Enterprise Console Setup** screen will load.
3. Ensure that a valid backup directory is defined. If the **Backup Directory** field is blank, type in a location relative to the `/pentaho/server/enterprise-console/` directory. **This must be an extant directory**; Enterprise Console cannot create directories on your filesystem. Click **OK** to save the configuration.
4. Go to the **Administration** section, then click the **Services** tab.
5. Click **Backup** in the **Solution Repository** section.  
Depending on how much content you have, this can take anywhere from a few seconds to several minutes. Click **Close** when the process is complete.

Your BA Server solution repository content and permissions are now backed up. To restore from a previous backup, click **Restore** and select from the list of backups.

## Checking the Status of the Solution Repository

---

To determine if your managed repository is responding to query requests...

1. In the **Pentaho Enterprise Console**, go to the **Status** page.
2. Under the **Repository** tab, in the **Information** group box, you see an item labeled "**Can the Pentaho Enterprise Console connect to the repository?**".  
If the value for this line is **true**, your repository is up and receiving query requests.

## Configuring the Solution Repository

---

You can change the server's repository configuration in the Pentaho Enterprise Console **Configuration** page. This configuration enables the server to communicate with the Managed Repository via Hibernate. For example, you may want to change the configuration so that the server can access information in a production database.

Changing the user name and/or password used to authenticate with the managed repository is likely the most frequent change to this configuration. You can, however, change other connection information, such as the JDBC driver class name, the Hibernate dialect to use for your repository, and the connection URL to the database.



**Caution:** Do not make changes to the Managed Repository configuration unless you are certain you know what you are doing. Improper changes to this configuration can cause your server to stop working.

To change this configuration setting...	Follow these instructions...
<b>Driver</b>	Enter a new driver class name in the text box and click <b>Submit</b> .
<b>User Id</b>	Enter a new user ID in the text box and click <b>Submit</b> .
<b>Connection Pool Size</b>	Enter a new connection pool size in the text box and click <b>Submit</b> .
<b>Dialect</b>	Enter a new dialect string in the text box and click <b>Submit</b> .
<b>URL</b>	Enter a new connection URL in the text box and click <b>Submit</b> .
<b>Change Password</b>	Enter the old database repository password in the <b>Old Password</b> text box. Enter the new database repository password in the <b>New Password</b> text box. Click <b>Submit</b> .

## Hibernate Dialects

Below is a list of dialects for each database type associated with the Managed Repository:

Database Type	Dialect
HSQLDB ("Hypersonic" — not for production use)	org.hibernate.dialect.HSQLDialect
MySQL	org.hibernate.dialect.MySQL5InnoDBDialect
Oracle 10g	org.hibernate.dialect.Oracle9Dialect
PostgreSQL	org.hibernate.dialect.PostgreSQLDialect

## Refreshing the Reporting Data Cache

---

By default, reports created with Report Designer and published to the BA Server will have their data sets cached to improve performance. If you are working with a volatile data source and will be regenerating reports frequently, you may need to manually flush the report cache, which will force the Reporting engine to recalculate the result sets for all reports.

To do this, log into the Pentaho User Console as an administrator user, then go to the **Tools** menu, select the **Refresh** sub-menu, and click the **Reporting Data Cache** item.

## Clearing the Content Repository

---

This procedure should only be performed during pre-announced maintenance periods. Clearing the content repository can cause undesired operation in BA Server content that is currently running; also, if a user executes content between the time the content repository is cleared and when the BA Server is shut down, the operation will have to be performed again.

The BA Server keeps its own cache for BA content artifacts. This makes certain content-related functions more efficient and responsive. However, there are some instances when you will want to clear out the content repository. Follow the directions below to clear the BA Server's content repository.

1. Ensure that there are no currently running BA Server schedules, and that the BA Server is running.
2. Log into the Pentaho Enterprise Console.
3. In Enterprise Console, go to the **Administration** section, then click on the **Services** tab. In the **Content Repository** pane, click **Execute**.
4. If you are unable or unwilling to use Enterprise Console, you can manually execute the **clean\_repository.xaction** action sequence in `/pentaho-solutions/admin/`.

The BA Server content repository has been cleared. New artifacts will be generated and stored as BA Server content is executed.

If you need to perform this operation on a regular and predictable basis, you can schedule the **clean\_repository.xaction** action sequence either through the Pentaho Enterprise Console or the Pentaho User Console.

## Clearing the Mondrian Cache

---

There is a default action sequence in the BA Server that will clear the Mondrian cache, which will force the cache to rebuild when a ROLAP schema is next accessed by the BA Server.

The cache-clearing action sequence is **clear\_mondrian\_schema\_cache.xaction**, and you can find it in the **admin** solution directory. This action sequence can be run directly from a URL by making the **admin** solution directory visible and then running the action sequence from the solution browser in the Pentaho User Console, from the Pentaho User Console by selecting the **Mondrian Schema Cache** entry in the **Refresh** part of the **Tools** menu, or by clicking the **Mondrian Cache** button in the **Administration** section of the Pentaho Enterprise Console:

```
http://localhost:8080/admin/clear_mondrian_schema_cache.xaction
```



## Content Promotion (Lifecycle Management)



**Danger:** Do not change the names of any content files, data sources, solution directories, or other file names during the promotion process. Names should be set during solution development and strictly maintained throughout content promotion. Modifying file names can result in complications with BA content that cannot be immediately detected, which will negatively impact your QA process.

This section explains supported processes for moving Pentaho content and server settings among multiple BA Server instances. Typically this includes either two or three disparate servers with identical configurations: One for BA content development, one (optionally) for testing and QA, and one for production.

In order to follow these instructions, you must have a fully-configured, feature-identical Pentaho environment for all servers that you want to move content to or from. There are no instructions in this section for setting up a BA Server instance; all necessary installation and post-install configuration material is covered earlier in this guide and in the installation documentation.



**Note:** You must have your own revision control system (CVS, Subversion, Git, etc.) in place before proceeding. The Pentaho lifecycle management process expects your company to have existing backup and revision control services that store production settings and data.

### Lifecycle Management Checklist

The Lifecycle Management Checklist is a basic plan intended to show one potential, recommended path for promoting BA Server settings, data sources, and content. This is not the only path to take, and it is not a complete set of step-by-step instructions. If you need more details than are provided in this checklist, consult the appropriate section in the verbose instruction set that comprises the rest of this section.



**Note:** If you find that this process is too technical for you to execute confidently on your own, you must enlist the help of an IT professional who is familiar with enterprise software lifecycle management. If there is no one in your organization who can assist you, you may be able to arrange for a Pentaho consultant to perform some or all of the preparation work.

Step	Procedure	Done
Step 1	Prepare your BA Server instances for lifecycle management by parameterizing the configuration files and committing the unified properties file to version control.	
Step 2	Ensure all data sources are able to be promoted (no Pentaho Data Source Wizard sources). Establish JNDI sources as replacements if necessary.	
Step 3	Shut down the originating BA Server and Enterprise Console (development or test).	
Step 4	Commit or archive the solution directories that you are promoting.	
Step 5	Create an SQL dump of the source hibernate database.	
Step 6	If you are promoting schedules, create an SQL dump of the source quartz database.	
Step 7	Clean all of the caches on the target server by using the maintenance functions in the Pentaho Enterprise Console, or by using a custom content repository and Mondrian schema cleanup script or xaction.	
Step 8	Physically transfer the solution directories, hibernate dump, and quartz dump files to the target machine (testing or production).	
Step 9	Start the source BA Server and notify users that it is back online.	
Step 10	Stop the destination BA Server, but not the solution database.	
Step 11	Physically transfer any relevant JNDI data source files to the target machine.	
Step 12	Import the hibernate SQL dump to the target machine.	
Step 13	If you are promoting schedules, import the quartz SQL dump to the target machine.	
Step 14	Import the hibernate SQL dump to the target machine.	

Step	Procedure	Done
Step 15	Start the destination BA Server and notify users that it is back online.	

## Preparing the Configuration for Promotion

### Objective

Configuration promotion is something that should happen rarely. Once a server is configured, it will probably not need to change unless there are performance tweaks or infrastructure changes. It is certainly not necessary to promote the configuration every time content or data sources are promoted.

The safest and easiest way to promote a configuration is to parameterize the necessary BA Server configuration files, then commit the corresponding properties files to revision control. Through this method, you can quickly and easily back up and restore only the required settings without touching content or software binaries.

### Discovery

In order to identify the files and settings that need to be parameterized, you will have to perform a diff operation between a fully-configured BA Server and an unconfigured one. If you don't have a diff program, investigate **Meld** (on Linux) or **Araxis** (Mac and Windows), or consult with your in-house development team for advice.

The most important BA Server configuration settings are stored in the `/server/biserver-ee/pentaho-solutions/system/` directory, with a few less critical files in `/server/biserver-ee/pentaho-solutions/admin/`. There are also a few core settings inside of the Pentaho WAR archive deployed to your application server, though they should not change at all after your initial server setup is complete.

To be absolutely certain that you have all of the BA Server configuration files selected, you should diff the following directories in their entirety:

- `/pentaho-solutions/system/`
- `/pentaho-solutions/admin/`
- `/WEB-INF/` inside your deployed pentaho.war
- `/META-INF/` inside your deployed pentaho.war
- For JBoss deployments, the PentahoHibernate-ds.xml and quartz-ds.xml files in the `/server/default/deploy/` directory



**Note:** There are binaries inside of the plugin directories for Analyzer, Dashboard Designer, Interactive Reporting, and Community Dashboard Framework. While it may be useful to take note any binary differences, which would indicate possible version differences among BA Server or plugin deployments, in general you should be most concerned with XML configuration files and properties files. However, if you have done any plugin customization work, you will need to promote those changes as well.

### Parameterization

Once you've identified all of your key configuration files, you should parameterize as many of them as you can so that there are fewer files to commit and promote. It's possible that you can narrow down the configuration promotion process to a single properties file, once all of your BA Server instances have been fully parameterized.

## Promoting Solutions

The Pentaho solution repository is a local directory that is mirrored in a relational database and persisted through Hibernate. In order to migrate a BA Server solution from one instance to another, you must copy the solution directory and export and import the Hibernate database. There are a variety of ways to do this; the procedure below explains one potential method.



**Danger:** You must copy the files **and** migrate the Hibernate database. If you do not copy the files, Hibernate will delete the database copies of them upon the next repository refresh operation (typically this happens at server startup, but it can be manually executed through the Pentaho User Console or via an action sequence).

1. Shut down the source BA Server and Pentaho Enterprise Console services (do not shut down the solution database, though).

```
/home/pentaho/pentaho/server/biserver-ee/stop-pentaho.sh
/home/pentaho/pentaho/server/enterprise-conole/stop-pec.sh
```

2. Use your preferred SQL query or database management tool to export the Hibernate database from the MySQL, PostgreSQL, or Oracle database that contains your solution repository.

```
mysqldump -u root -p hibernate > new_solutions.sql
```

3. Commit your solution directory or directories to version control, or compress them into a single archive in anticipation of transferring them to the destination machine.

```
tar -cpzf new-solution1.tar.gz /home/pentaho-devel/pentaho/server/biserver-ee/
pentaho-solutions/new-solution1
```

4. On the destination machine, update the working copy of your solution directories from version control, or unpack the solution archive.
5. On the destination machine, execute the hibernate SQL dump.

Your new solution has been moved from the previous BA Server instance to the new one.

## Migrating Data Sources

### Database connections

If you are using Pentaho managed data sources (created with the Data Source Wizard), you will have to recreate them as traditional JNDI data sources, which are easily ported to other application servers. Examine each Data Source Wizard source and transfer the connection details to a Tomcat- or JBoss-based data source definition.

If you are using JBoss, select your **\*ds.xml** JNDI data source files for promotion, as well as the `/jboss/server/default/deploy/pentaho.war/WEB-INF/web.xml` and `/jboss/server/default/deploy/pentaho.war/WEB-INF/jboss-web.xml` files.

If you are using Tomcat, select the `/tomcat/conf/web.xml` and either the global `/tomcat/conf/context.xml` or the local `/tomcat/webapps/pentaho/WEB-INF/context.xml` file for promotion. If you've defined your JNDI connections in `/tomcat/conf/server.xml`, select that in addition to or instead of `context.xml`.

If you are upgrading your database drivers as part of your data source promotion, ensure that the updated JARs are included in your procedure. In JBoss, they are stored in `<jboss>/server/<server config>/lib/` (though there are other lib directories that may also apply to your configuration). In Tomcat, they are stored in `/tomcat/lib/`. Also ensure that driver JARs in client tools, the DI Server, and the `pentaho.war` itself are identical.

### Metadata definitions

Pentaho Metadata XML data sources are stored in individual solution directories, so they will be promoted along with the solution content files.

### ROLAP schemas

Published Mondrian schemas are stored in individual solution directories, so they will be promoted along with the solution content files. However, there is a list of published schemas in the `/pentaho-solutions/system/olap/datasources.xml` that you should select for promotion. This file populates the data source list in Pentaho Analyzer; without it, published schemas will not be available to Analyzer users.

## Promoting Schedules

You can only migrate schedules **after** solution content and data sources have been promoted. Schedule promotion is the last step in the content promotion process.

Pentaho content schedules are stored in a relational database called **quartz**. If predefined schedules are part of your solution development and/or testing processes, quartz must be included in the promotion procedure.

1. Shut down the source BA Server and Pentaho Enterprise Console services (do not shut down the schedule database, though).

```
/home/pentaho-devel/pentaho/server/biserver-ee/stop-pentaho.sh
/home/pentaho-devel/pentaho/server/enterprise-conole/stop-pec.sh
```

2. Use your preferred SQL query or database management tool to export the **quartz** database from the MySQL, PostgreSQL, or Oracle database that contains your schedules.

```
mysqldump -u root -p quartz > new_schedules.sql
```

3. Transfer the SQL backup file to the destination server.

```
scp new_schedules.sql pentaho-test@gamachine_02:
```

4. Execute the SQL script to overwrite the current quartz database on the destination machine.

```
mysql -u root -p < new_schedules.sql
```

Your new schedules have been moved from the previous BA Server instance to the new one.

## Artifact Cleanup

---

The BA Server generates certain temporary files and caches to improve the performance of content generation, scheduling, and delivery.

All destination server cleanup should happen during a maintenance interval when no users or schedules are executing content. This should be the last step before shutting down the destination server for content promotion from development or testing.

You should already have a plan or schedule for clearing the BA Server's content repository on a regular basis. However, whenever you promote content to a server, you must first force the cleanup to happen on the destination machine. If you don't have a custom script or action sequence that clears the content repository, you can manually clear it: [Clearing the Content Repository](#) on page 40.

The Mondrian cache must be cleared prior to promoting modified ROLAP schemas. If you don't already have a process in place for this, follow these directions: [Clearing the Mondrian Cache](#) on page 40.

# Logging, Performance Monitoring, and Troubleshooting

This section provides you with instructions for diagnosing BA server errors and information about how to use the server performance monitoring tools.

## Viewing Errors

To view the list of BA Server warnings and errors...

1. In the Pentaho Enterprise Console home page, click **Status**.
2. Click on the **Errors** tab to display all the errors that the Pentaho Enterprise Console has identified.
3. To resolve the individual errors, go to the specific location in the console from which the warning or error originated.

## Viewing Environment Settings

Environment settings provide you with detailed information about about your "environment" such as the operating system you are using, the Java version you are using, the address of your Tomcat server, and more.

To view the Environment Settings...

1. In the Pentaho Enterprise Console home page, click **Status**.
2. Click on the **Environment** tab to display the Java Virtual Machine (JVM) system settings of the application server on which the BA Server is running



**Note:** If the application server is not configured or not online these values are not available.

## Viewing Security Settings

Many of the security settings are distributed across parts of the Pentaho Enterprise Console:



**Note:** The *Pentaho Security Guide* has detailed instructions for configuring BA Server and Pentaho Enterprise Console security.

Setting	Description
<b>Access Control</b>	To view and modify the Access Control settings, in the Pentaho Enterprise Console home page, click <b>Configuration</b> , then click on the <b>General</b> tab. The Access Control group box allows you to customize <b>Access Control Lists</b> (ACLs) and other low level access control settings.
<b>Authentication</b>	To view and modify the <b>Authentication</b> setting, in the Pentaho Enterprise Console home page, click <b>Configuration</b> , then click on the <b>Web Settings</b> tab. The current authentication options include memory-based, RDBMS-based, and LDAP-based.
<b>LDAP Configuration</b>	To view and modify the LDAP configuration, in the Pentaho Enterprise Console home page, click <b>Utilities</b> , then click on the <b>LDAP Configuration</b> tab. Refer to the <i>Pentaho BA Platform Security Guide</i> for additional documentation about the LDAP Configuration.

## LDAP

The Pentaho Enterprise Console has an LDAP section in the **Utilities** category. The fields in this screen should be relatively intuitive, but are explained below in more detail in case there are options that you are not familiar with:

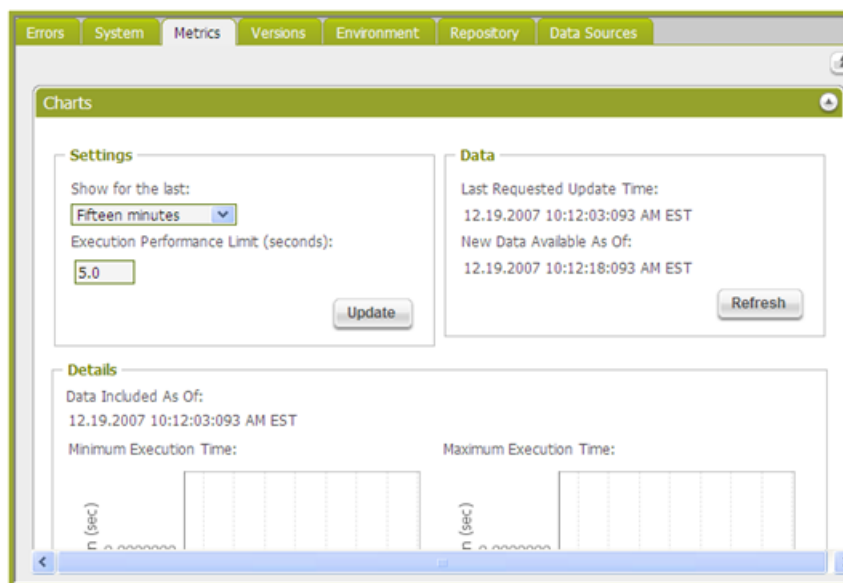
Property Group Name	Description
<b>Directory Connection</b>	Properties used for connecting to an LDAP directory (such as the host name).
<b>User Search</b>	Properties that describe a query that gets a user's record using his or her user name. When users attempt to log on they provide a user name and password. The BA Server uses the user name given at logon, along with the properties in this group, to get the user's record.
<b>Authorities Search</b>	Properties that describe a query that gets all roles that should be recognized by the BA Server. The BA Server needs a complete list of roles for the purpose of exposing them in an interface used to assign permissions to action sequences. (This is the "Permissions" interface.)
<b>Populator</b>	Properties that describe a query that gets a user's roles using his/her user name or Distinguished Name (DN). Because there is no standard for where to store roles in a directory, roles could be stored anywhere in the directory tree. To support this inconsistency, the BA Server provides a separate search.


## Monitoring Performance

The Metrics page associated with the **Status** menu item provides key performance metrics for your BA Server. The metrics provided give you insight into the latest action sequence execution activity on your server, the latest user activity, and action sequence execution analysis over a specified period of time.

### Performance Metrics Values

The atomic performance values are all based on the **Show for last (minutes)** interval (with the exception of the Logged In Users metric). Each metric is calculated relative to this interval, which is the number of minutes from the time the metrics were last refreshed. For example, if your interval is set to 15 minutes, the Average Duration metric displays the average execution time for all action sequences executed in the last 15 minutes ("last" meaning the time the metrics were last refreshed). The performance metrics are calculated using data gathered from the BA Server Enterprise Edition RDBMS audit logging engine.



 **Note:** By default, the BA Server is configured to log audit information to an RDBMS. If you are not using the default configuration, performance metrics may not be available.

The table below provides a description of what each metric measures:

Metric	Description
<b>Average Duration</b>	The average execution time for all action sequences executed in the last metrics interval.
<b>Maximum Duration</b>	The maximum execution time for all action sequences executed in the last metrics interval.
<b>Minimum Duration</b>	The minimum execution time for all action sequences executed in the last metrics interval.
<b>Number of Executions</b>	The number of action sequences executed in the last metrics interval.
<b>(Users) Current Active/Logged in</b>	The number of users who have completed execution of at least one action sequence in the last metrics interval. Note that this user may no longer be logged on, or have an active session. The only thing this metric indicates is that the user performed some activity within the last interval.
<b>Number of Logged In Users</b>	This metric displays the current number of authenticated users that are logged on. Users are added to the count of logged on users when they log onto the server, and are removed when their session times out or they manually log off. This metric does not include users who anonymously access the server. This value is refreshed only when the performance metrics are refreshed manually.

## Changing the Show for Last Interval

To change the **Show for Last interval** to a more relevant value for your environment...

1. Enter a new value in the **Show for Last** text box. This value can be any decimal number and represents the interval in **minutes**.
2. Click **Update**.  
Changing the Show for Last value updates the performance metrics.

## Refreshing the Metrics Values

Metrics values must be refreshed manually. There is no automatic update of the metrics values, except at start up of the Pentaho Enterprise Console. To refresh metric values, click **Submit**. Each metric is updated, and the **Last Updated** value is updated to reflect the latest refresh date and time.

The screenshot shows the 'Metrics' tab in the Pentaho Enterprise Console. The 'Usage' section displays the 'Show for last (minutes)' as 15.0 and the 'Last Updated' time as 12.19.2007 10:09:46:812 AM EST. Below this, the 'Action Sequence' section shows Average Duration (seconds) as 0.0, Maximum Duration (seconds) as 0.0, Minimum Duration (seconds) as 0.0, and Number of Executions as 0. The 'Users' section shows Current Active as 0 and Logged In as 0. A 'submit' button is located at the bottom right of the metrics display area.



**Note:** Refreshing the performance metrics values does not refresh the performance charts. See [Refreshing the Performance Charts](#) for more information on updating the performance charts.



## Preventing Timeout Transaction Errors Associated with Scheduled Reports

When you create a schedule to which you add multiple reports, it's possible to add so many reports that the Subscription (Public Schedule) crashes. The crash occurs because the Hibernate Managed Subscription is unable to complete each transaction in under 50 seconds. The result includes state exceptions, and other problems, on subsequent runs that eventually lead to a Java heap space out-of-memory error. The issue occurs when running MySQL.

Follow the instructions below correct the problem:

1. Stop the BA server and MySQL, consecutively.
2. Using a text editor, open the MySQL **my.ini** file (in Windows) or the **my.cnf** file (in Linux/MAC).
3. Locate the following line: **innodb\_lock\_wait\_timeout = 50**

If you don't see this parameter, add it to the file.

4. Change the timeout from 50 to a higher number that can handle the load for a single (highly loaded) public schedule and save the file.



### Note:

You may need to adjust the timeout value to find the right threshold. Setting it way too high is not recommended because in instances where a transaction cannot complete because of an error, the thread hangs for the timeout length.

5. Restart the BA server and MySQL, consecutively.

## Using Performance Charts

The performance charts are all based on the **Show for the last** period selected. The time periods available to choose from are:

- Fifteen minutes
- Thirty minutes
- One hour
- Twelve hours
- 24 hours
- Seven days
- Thirty days
- Ninety days

Each chart plots data available from the BA Server audit logging engine. Data is selected that falls within the selected time period, and is then aggregated and/or calculated according to that chart's purpose. For example, if you select the "One hour" time period, then the charts display the trend for action sequence executions over the last hour ("last" meaning the time the charts were last refreshed).

Chart Type	Description
<b>Minimum Execution Time Chart</b>	Displays the minimum amount of time it took for action sequences to execute over the Analysis Time Period.
<b>Maximum Execution Time Chart</b>	Displays the maximum amount of time it took for action sequences to execute over the Analysis Time Period.
<b>Number of Executions Chart</b>	Displays how many action sequences executed in intervals over the Analysis Time Period. Execution Performance Chart
<b>Execution Performance Chart</b>	Displays the percentage of action sequences (on the Y axis) that executed within a given number of seconds (the x axis) over the Analysis Time Period. This is a cumulative chart, so all action sequences that took less than one second also took less than 2 seconds, and less than 3 seconds, and so on.



Chart Type	Description
<b>Execution Performance Limit Deviation Chart</b>	Displays the percentage of action sequences that executed successfully within a selected Execution Performance Limit. This limit, expressed in seconds, is configurable. See <a href="#">Changing the Execution Performance Limit for the Execution Performance Limit Deviation Chart</a>

### Changing the Chart Analysis Time Period

To change the analysis time period to a more suitable value that makes sense in your environment...

1. Select a new value in the **Show for the last** selection list.
2. Click **Submit**.

### Changing the Execution Performance Limit for the Execution Performance Limit Deviation Chart

Once you have analyzed your execution data, you will have a better idea of what your acceptable performance limit for action sequence execution is. At that point, you may want to change the **Execution Performance Limit**.

1. Enter a new value in the **Execution Performance Limit** (seconds) text box. This value can be any decimal number, and represents the limit in seconds.
2. Click **Submit**.  
Changing the Execution Performance Limit updates the performance charts.

### Refreshing the Performance Charts

Currently, there is no automatic update of the performance charts, except at start up of the Pentaho Enterprise Console. To manually update the charts, under **Data**, click **Refresh**.

All charts are updated, and the **Last Requested Update Time** is updated to reflect the latest refresh date and time. The **Data Included As Of** date is updated. The Data Included As Of date value represents the date and time limit for data included in the chart. This differs from the Last Requested Update Time due to calculating the chart data intervals over multiple updates to the charts and attempting to keep the data displayed consistently from one update to the next. And last, the **Data Available As Of** date value is updated, which reflects the next date and time new data will be available to the charts. If you attempt to update the charts before this date and time, the charts will not update, as new data is not yet available.



**Note:** This availability is based on the analysis time period. By changing the analysis time period, you essentially reset these dates, and the charts update immediately, regardless of the value of the **Data Available As Of** date value.



**Note:** Refreshing the performance charts does not refresh the performance metrics values. See [Refreshing the Metrics Values](#) for more information on updating the performance metrics values.

## Estimating the Size of the Audit Log

Turning on audit logging means that the size of your solution repository will grow at a faster rate. Some DBAs may need to accommodate for this accelerated growth, so this section explains how to calculate audit log growth.

First of all, the size of the audit log does not depend on the logging level used.

The average length of an audit entry is 220 characters. Each of the following events is logged:

- Session start
- Action sequence execution start
- BA component execution start
- BA component execution end
- Action sequence execution end

An action sequence with four actions defined in it will generate about 2200 characters in the audit log.  $2200 = 220 \times 10$ . The 10 comes from 2 (for the xaction start and end) +  $(2 \times 4)$  (for each component).

If the DB-based audit option is used, the raw DB-based audit entries will be the same size or slightly smaller.

The raw DB tables or log files are transformed via ETL into the audit reporting tables. These tables are smaller than the raw tables as the start and end rows are collapsed into single rows, so these tables will be about half the size of the raw tables.

The raw log files or tables do not have to be kept once they have been processed into the audit reporting tables.

Given these variables:

- **U** = average number of users per 24-hour period
- **R** = average number of reports (xactions) run per user per day
- **C** = average number of components per xaction

Approx size of raw logs (file or DB) =  $U \times R \times (440 + C \times 440)$

Approx size of audit reporting tables =  $U \times R \times (220 + C \times 220)$

A slightly simpler version of the calcs:

Approx size of raw logs (file or DB) =  $U \times R \times (C+1) \times 440$

Approx size of audit reporting tables =  $U \times R \times (C+1) \times 220$

Another way to look at it is an xaction with 3 components will add, per execution, about 1.7k of raw log and 0.86k of audit reporting data.

## Switching to File-Based Audit Logging

In high-use environments, performance issues may occur due to a sizeable number of SQL inserts associated with audit logging. In this instance, you can switch to file-based audit logging.

1. Open the file, **pentahoObjects.spring.xml**, located at `...\pentaho-solutions\system`.
2. Locate the following line:

```
<bean id="IAuditEntry"
 class="org.pentaho.platform.engine.services.audit.AuditSQLEntry"
```

```
scope="singleton" />
```

3. Edit the line as follows:

```
<bean id="IAuditEntry"
 class="org.pentaho.platform.engine.services.audit.AuditFileEntry"
 scope="singleton" />
```

4. Save the file and restart the BA Server.


### Removing Audit Logging

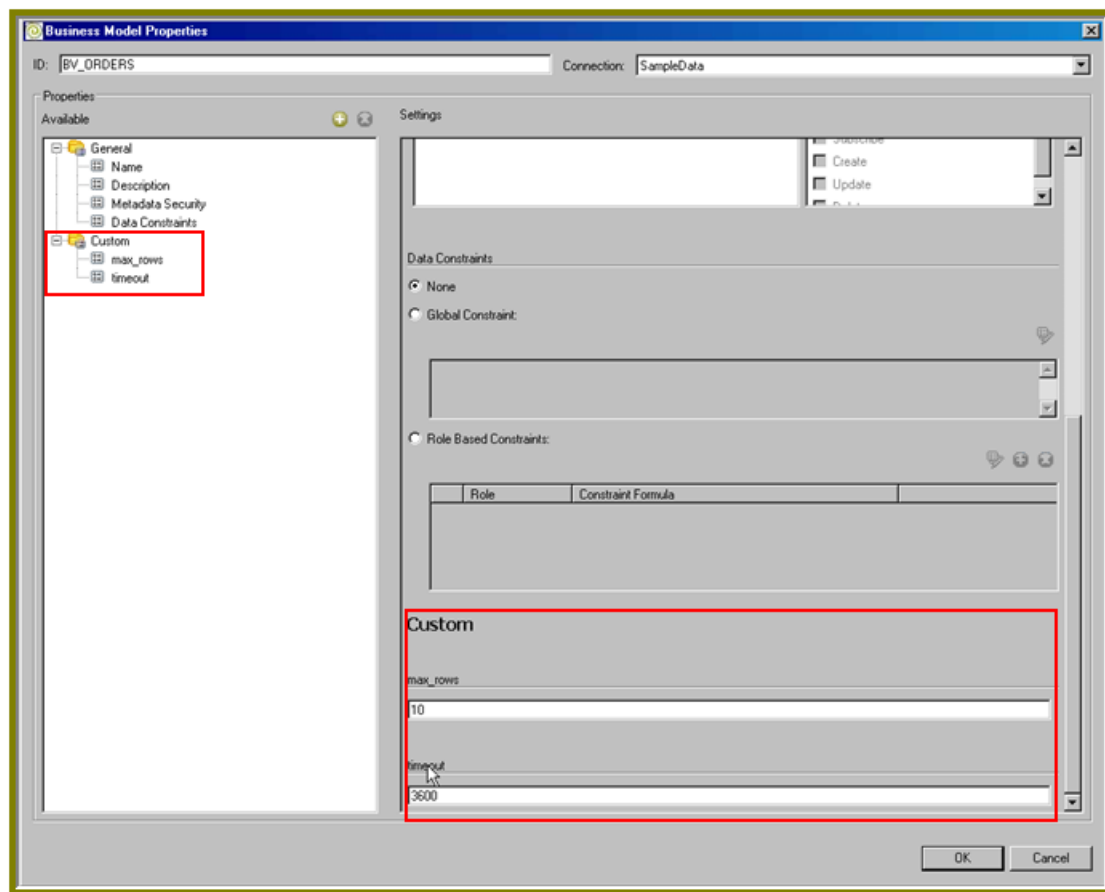
Follow the instructions below if you do not want to use audit logging. You may want to disable audit logging when you are in a test or development environment. Disabling audit logging improves performance; however, make sure you have no need to evaluate data contained in audit logs.

1. Go to ... \pentaho-solutions\system\.
2. Open the **pentahoObjects.spring.xml** file
3. Locate the line: `<bean id="IAuditEntry"`  
`class="org.pentaho.platform.engine.services.audit.AuditSQLEntry" scope="singleton" />`
4. Replace the line above with this line: `<bean id="IAuditEntry"`  
`class="org.pentaho.platform.engine.core.audit.NullAuditEntry" scope="singleton" />`
5. Save the file and restart the BA Server.


### Defining Result Row Limit and Timeout


When a Web-based Ad Hoc query in the Pentaho User Console returns an unusually large number of rows, this may impact server performance. To limit the number of rows returned by a query and to set up a timeout, you must create two custom properties, **max\_rows** and **timeout**, in the Pentaho Metadata Editor.

 **Note:** The values you define for the row number limit (max-rows) and timeout properties are passed to the JDBC driver.



To define max rows and timeout...

1. In the Pentaho Metadata Editor, expand the **Business Model** node and select **Orders**.
2. Right-click Orders and choose **Edit**.  
The **Business Model Properties** page displays a list of properties that were previously defined.
3. In the Business Model Properties page, click  (Add).  
The Add New Property page dialog box appears.
4. Enable **Add a custom property**.
5. In the **ID** text box, type **max\_rows**.

 **Important:** The ID is case-sensitive and must be typed exactly as shown.


6. Click the down-arrow in the **Type** field and choose, **Numeric**.  
The **Business Model Properties** page appears. The **max\_rows** property is listed under **Custom** in the navigation tree.
7. In the right pane, under **Custom**, enter a value for your **max\_rows** property. For example, if you enter "3000" as your value, the number of rows allowed to display in a query result is constrained to 3,000.
8. Repeat steps 3 through 6 to for the **timeout** custom property.
9. In the right pane, under **Custom**, enter a value for your **timeout** property. The timeout property requires a numeric value defined in number of seconds. For example, if you enter, 3600, the limit for query results is one minute.
10. Click **OK** in the Business Model Properties page to save your newly created properties.

## Column Metadata Security Restrictions On Columns With Inner Joins

You may experience malformed SQL query results when creating ad hoc reports if your metadata model contains security restrictions on columns that are joined to other columns. This is not a supported action in Pentaho Metadata's security infrastructure. An error message will appear that explains this limitation.

## Disabling Server and Session-Related Timeouts in the Pentaho User Console

Follow the instructions below to disable server and session timeouts associated with the **Pentaho User Console**.

 **Important:** These instructions are applicable when you are in a *test* environment. Once you go live, it is recommended that you set your timeouts to five or ten minutes so that sensitive BA Server-related data can be protected. The time must be expressed in minutes.

1. Open the file, **server.xml**, located under ...\\biserver-ee\\tomcat\\conf.
2. Find the **connectionTimeout="20000"** parameter and change its value to zero ("**0**"). If this value is set to a negative number it will never timeout.
3. Open the file **web.xml**, located under ... \\bi-server\_ee\\tomcat\\webapps\\penaho\\WEB-INF\\web.xml.
4. Find the **session-timeout** parameter and changes its value to negative one ("**-1**").
5. Save the file and refresh the user console.

## Log Rotation

This procedure assumes that you do not have or do not want to use an operating system-level log rotation service. If you are using such a service on your Pentaho server, you should probably connect it to the Enterprise Console, BA Server, and DI Server and use that instead of implementing the solution below.

Enterprise Console and the BA and DI servers use the Apache log4j Java logging framework to store server feedback. The default settings in the log4j.xml configuration file may be too verbose and grow too large for some production environments. Follow the instructions below to modify the settings so that Pentaho server log files are rotated and compressed.

1. Stop all relevant Pentaho servers -- BA Server, DI Server, Pentaho Enterprise Console.
2. Download a Zip archive of the Apache Extras Companion for log4j package: <http://logging.apache.org/log4j/companions/extras/>.
3. Unpack the **apache-log4j-extras** JAR file from the Zip archive, and copy it to the following locations:

- **BA Server:** /tomcat/webapps/pentaho/WEB-INF/lib/
  - **DI Server:** /tomcat/webapps/pentaho-di/WEB-INF/lib/
  - **Enterprise Console:** /enterprise-console/lib/
4. Edit the **log4j.xml** settings file for each server that you are configuring. The files are in the following locations:
    - **BA Server:** /tomcat/webapps/pentaho/WEB-INF/classes/
    - **DI Server:** /tomcat/webapps/pentaho-di/WEB-INF/classes/
    - **Enterprise Console:** /enterprise-console/resource/config/
  5. Remove all **PENTAHOCONSOLE** appenders from the configuration.
  6. Modify the **PENTAHOFILE** appenders to match the log rotation conditions that you prefer.

You may need to consult the log4j documentation to learn more about configuration options. Two examples that many Pentaho customers find useful are listed below:

#### Daily (date-based) log rotation with compression:

```
<appender name="PENTAHOFILE" class="org.apache.log4j.rolling.RollingFileAppender">
 <!-- The active file to log to; this example is for BA/DI Server.
 Enterprise console would be "server.log" -->
 <param name="File" value="../../../logs/pentaho.log" />
 <param name="Append" value="false" />
 <rollingPolicy class="org.apache.log4j.rolling.TimeBasedRollingPolicy">
 <!-- See javadoc for TimeBasedRollingPolicy -->
 <!-- Change this value to "server.%d.log.gz" for PEC -->
 <param name="FileNamePattern" value="../../../logs/pentaho.%d.log.gz" />
 </rollingPolicy>
 <layout class="org.apache.log4j.PatternLayout">
 <param name="ConversionPattern" value="%d %-5p [%c] %m%n" />
 </layout>
</appender>
```

#### Size-based log rotation with compression:

```
<appender name="PENTAHOFILE" class="org.apache.log4j.rolling.RollingFileAppender">
 <!-- The active file to log to; this example is for BA/DI Server.
 Enterprise console would be "server.log" -->
 <param name="File" value="../../../logs/pentaho.log" />
 <param name="Append" value="false" />
 <rollingPolicy class="org.apache.log4j.rolling.FixedWindowRollingPolicy">
 <!-- Change this value to "server.%i.log.gz" for PEC -->
 <param name="FileNamePattern" value="../../../logs/pentaho.%i.log.gz" />
 <param name="maxIndex" value="10" />
 <param name="minIndex" value="1" />
 </rollingPolicy>
 <triggeringPolicy class="org.apache.log4j.rolling.SizeBasedTriggeringPolicy">
 <!-- size in bytes -->
 <param name="MaxFileSize" value="10000000" />
 </triggeringPolicy>
 <layout class="org.apache.log4j.PatternLayout">
 <param name="ConversionPattern" value="%d %-5p [%c] %m%n" />
 </layout>
</appender>
```

7. Save and close the file, then start all affected servers to test the configuration.

You now have an independent log rotation system in place for all modified Pentaho servers.

## Troubleshooting

This section contains known problems and solutions relating to the procedures covered in this guide.

## Address Already In Use: JVM\_Bind When Starting Enterprise Console

If you encounter this error when starting Enterprise Console, then the default port 8088 is not available on your system. To change the Enterprise Console port, edit the `/pentaho/server/enterprise-console/resource/config/console.properties` file and change the value of **console.start.port.number** to an open port.

## Case-Insensitivity for Usernames, Passwords, and Filenames

This problem can manifest in two different known ways. It can show as case-insensitivity as mentioned above, or it can show up when you change the case of an action sequence file name.

The source of the problem is the collation method and character set for your server. Selecting the wrong configuration will cause all usernames, passwords, and content files stored in the database to become identical in terms of characters without regard for case; this can cause name collisions. SUZY, Suzy, and suzy will all be the same username as far as the BA Server will be able to tell. To change this, you must set your database or table collation method to **latin1\_general\_cs**. You can do this by editing the database scripts mentioned below, or by modifying your server configuration.

## HTTP 500 or "Unable to connect to BA Server" Errors When Trying to Access Enterprise Console

If you have trouble controlling the BA Server through the Pentaho Enterprise Console, it is likely because you've changed the BA Server's IP address or hostname (the **fully-qualified-server-url** node in `web.xml`) from the default 127.0.0.1 to something else without also applying this change to the Proxy Trusting Filter in the `web.xml` file. See [Configuring the Proxy Trusting Filter](#) on page 11 for instructions on how to fix this.

## Windows Domains Won't Authenticate When Using the JTDS Driver

If you are using a JTDS JDBC driver and you want to use a Windows domain user to authenticate to a Microsoft SQL Server, be aware that the Windows syntax will not work for specifying the domain and user. In the URL field, the domain must be appended to the end of the URL with a semicolon:

```
jdbc:jtds:sqlserver://svn-devel.example.com:1533/reportsInProgress;domain=testdomain
```

## Geo Maps partially rendered

If your Geo Map visualizations in Analyzer are not correctly displaying, it's possible that the BA Server is not giving them enough time to fully render. See [Setting the Geo Map Rendering Wait Time](#) on page 35 for more information.

## Getting Support

---

The Support menu item contains four group boxes that provide you with assistance when a BA server problem arises:

- The **Export Settings** group box allows you to create an XML file that contains your BA server settings and configuration to send to the Pentaho support team. Support may request that you send this file when they need to diagnose an issue. To export your configuration and settings to an XML file, click **Export to XML**.
- The **Links** group box contains links to helpful articles that may help you address a BA server-related issue.
- The **FAQ** group box contains questions and answers to common issues associated with your BA server.
- The **How Tos** group box contains updates for issues you submit to support.

## Working From the Command Line Interface

---

Though the Pentaho Enterprise Console is the quickest, easiest, and most comprehensive way to manage PDI and/or the BA Server, some Pentaho customers may be in environments where it is difficult or impossible to deploy or use the console. See the alternative instructions for command line interface (CLI) license registration for step-by-step instructions.

### Managing Enterprise Edition Keys (CLI)

---

To list or remove Pentaho Enterprise Edition keys in the command line interface, follow the below instructions.

1. Navigate to the `/pentaho/server/biserver-ee/enterprise-console/license-installer/` directory.
2. Run the **install\_license.bat** (on Windows) or **install\_license.sh** (on Linux) script with the **display** switch.

```
install_license.sh display
```

If you have installed any Enterprise Edition keys, a list of them will appear, along with the products they cover and the term of the support contract.

3. To remove a license, run the **install\_license** script with the **uninstall** switch.

```
install_license.bat uninstall
```

A list of installed licenses will appear, followed by a prompt for the license ID you would like to remove. If you press Enter at this prompt, it will exit without taking any action.

4. Type in the license ID number that you want to remove, then press Enter.

After removing a key, if you had more than one installed, the list will regenerate and the prompt will reappear. You can choose to remove another Enterprise Edition key, or you can press Enter to exit the script.



**Note:** If you would prefer not to be prompted for confirmation, or if you intend to call this program as part of a script, use the **-q** switch to suppress prompting.

### Resetting the Default Access Control Lists (CLI)

---

Follow the below process to reset the default ACLs from the command line interface.

1. Stop the Pentaho BA Platform service or server.



**Note:** The solution database server must remain running for this procedure.

```
sh /usr/local/pentaho/server/biserver-ee/stop-pentaho.sh
```

2. Execute the following SQL statements using your database's command line utility:  
**mysql**, **psql**, and **sqlplus** are the preferred utilities for MySQL, PostgreSQL, and Oracle, respectively.

```
DROP TABLE PRO_ACLS_LIST;
DROP TABLE PRO_FILES;
```

3. Start the Pentaho BA Platform service or server.

```
sh /usr/local/pentaho/server/biserver-ee/start-pentaho.sh
```

The dropped tables have been recreated with default values.