Pentaho Archive-Based Installation Guide

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

Archive-Based Installation Overview

Pentaho provides several installation paths to address a variety of computing environment requirements. These instructions show you how to install each Pentaho product with the specific databases and software you already have or prefer to supply for yourself. Typically, the Business Analytics Server and Pentaho Enterprise Console will go on the server, and each workstation will have its own copy of the plugins.

Use this guide if you...

- Have your own or are creating your own deployment scripts
- Are using a command line interface or are performing a headless installation
- Are performing a remote installation
- Plan to connect to a database that is not the default installation database in the installation wizard
- Are independently upgrading or migrating to a new version of a particular product or plugin
- Need more fine-grained control over configuration and deployment options for servers

If you would rather ...

- Use your own application server
- Implement single sign-on through a Central Authentication Service

Use the *Pentaho BA Server Manual Deployment Guide*.

If you want all of this functionality...

- Install using a graphical installer
- Install quickly and/or for evaluation purposes
- Use the default PostgreSQL database
- Use the Tomcat application server

Use the *Pentaho Business Analytics Graphical Installer Guide*.

If you need to install the Pentaho Data Integration server, refer to the *Pentaho Data Integration Installation Guide*.

To ensure that your installation goes smoothly, refer to the *Compatability Matrix: Supported Components* for details about the versions of databases, application servers, software, and browsers that are compatible with Pentaho Business Analytics Enterprise Edition.

Components

This table shows the components included in the graphical installation and the databases and software you must supply on your own.

Included in Pentaho Archive Packages	You Must Supply
Application Server: • Tomcat **Business Analytics Server**: • Pentaho User Console • Analyzer • Dashboard Designer • Interactive Reporting • Pentaho Enterprise Console • Pentaho Mobile **Design Tools:** • Schema Workbench • Pentaho Data Integration • Metadata Editor • Report Designer • Design Studio	**A supported operating system:** • Linux • Solaris • Windows • Mac OS **The Sun JRE** Installed on the server and each workstation **A Supported Solution Database:** • MySQL • PostgreSQL • Oracle **One or More Data Sources:** • Any JDBC-compliant database • A spreadsheet • A flat file containing comma-separated values

Included in Pentaho Archive Packages	You Must Supply
• Aggregation Designer	

Prerequisites

In order to install Pentaho Business Analytics, you must be very familiar with technical concepts.

- You must be familiar with system administration operations pertaining to network services, including modifying your firewall to open specific ports, and adding services to the system startup and shutdown scripts.
- You must feel comfortable using the operating system command line interface and/or graphical system administration tools.
- You or a system administrator must have the ability to install software, open firewall ports, and start and stop system services on the machine you are installing on.

Installing Pentaho Business Analytics client tools onto end-user workstations is a much simpler process which requires running the graphical installation utility and customizing it slightly to fit your needs.

Installation Checklist

The Installation Checklist is a concise list of instructions intended to show a high-level overview of the archive-based installation and configuration process. It also serves as a quick reference for administrators and developers who have performed several installations in the past and only need a brief rehash of the necessary steps. **This is not the complete instruction set.** 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 guide.

Step	Procedure	Done
	Download archive packages for the BA Server, Pentaho Enterprise Console, and any BA Server plugins that you have licenses for from the Pentaho Customer Support Portal. See *Obtaining the Installation Materials* for more information.	
	For Unix Operating Systems, create a Pentaho System User. See *Creating a Pentaho System User in Unix Operating Systems* and *Local User Accounts for Unix Operating Systems* for more information.	
	Establish a **PENTAHO_JAVA_HOME** system variable that points to a Sun JRE or JDK. See *Setting the PENTAHO_JAVA_HOME Variable on Linux* or *Setting the PENTAHO_JAVA_HOME Variable on Windows* for more information.	
	Create a `/pentaho/server/` directory in a location accessible to the user account that will run the application server (the `/home/pentaho/` directory on Linux), then unpack the BA Server and Pentaho Enterprise Console archives to it. See *Deploying the BA Server* for more information.	
	Copy the Solution Database JDBC Driver JAR. See *Copying Solution Database JDBC Drivers* for more information.	
	Using the included SQL scripts for your database of choice, initialize your Pentaho solution repository. See *Initializing a MySQL, PostgreSQL, or Oracle Database* for more information.	
	For Oracle or MySQL solution databases, modify the `context.xml, pentaho.xml, hibernate-settings.xml, and applicationContext-spring-security-hibernate.properties` files appropriately. See *Initializing a MySQL or Oracle Database*.	
	Optionally, modify the appropriate XML configuration files to change the default hostname from **localhost**. See *Changing the Solution Database Hostname* for more information.	
	Optionally, modify Tomcat's **server.xml** to change the BA Server and Pentaho Enterprise Console port numbers. See *Changing the Default Port Numbers* for more information.	
	If you have Pentaho Dashboards Enterprise Edition, Pentaho Interactive Reporting, Pentaho Mobile or Pentaho Analysis Enterprise Edition, install the product plugins by	

Step	Procedure	Done
	unpacking them to `/pentaho/server/biserver-ee/pentaho- solutions/system/`.	
	If you have a headless CentOS, Red Hat, or Ubuntu environment, you may wish to configure dashboard printing. See *Enabling Dashboard Printing on Headless CentOS and Red Hat* or *Enabling Dashboard Printing on a Headless Ubuntu* for more information.	
	If you want install the Analysis Enterprise Edition Plugin, see *Installing the Analysis Enterprise Edition Plugin* and *Switching to Memcached*.	
	If you want to add support for Analyzer Geo Maps, see *Adding Linux and OS X Support for Printing Analyzer Geo Maps* or *Adding Windows Support for Printing Analzyer Geo Maps*.	
	Start the BA Server, then start the Pentaho Enterprise Console (you must be logged in as the **pentaho** user to do this if you are on Linux or Solaris) by running the `/pentaho/server/biserver-ee/start-pentaho` and `/pentaho/server/enterprise-console/start-pec` scripts, respectively. See *Initial Startup of the BA Server and Pentaho Enterprise Console* for more information.	
	Log into the Pentaho Enterprise Console at `http://localhost:8088`, or whatever host and port you configured, then install licenses. Establish users, roles, and data sources. Perform any other necessary configuration tasks. See *Post-Install Configuration* for more information.	
	Optionally, modify your Linux or Windows startup configuration so that the BA Server and Pentaho Enterprise Console will start automatically at boot time. See *Starting the BA Server At Boot Time On ...* for your operating system for more information.	
	On each client workstation, create an accessible `/pentaho/design-tools/` directory, then download the archive packages for the Pentaho client tools and unpack them to that directory. See *Installing the Pentaho Client Tools* for more information.	
	Access the BA Server from a workstation and verify that you can log in as a normal user, and create reports, analysis views, and dashboards.	
	When you are ready to move this server to development or production, remove the Pentaho sample data and solutions. You can also switch to a production login screen. See *Removing Pentaho Sample Data and Solutions* and *Switching to a Production Login Screen*.	
	Refer to *Compatibility Matrix: Supported Components* to verify version numbers.	
	Refer to the *Troubleshooting* section if you need additional help.	

Obtaining the Installation Materials

As an Enterprise Edition customer, you can obtain the BA Server and client tool pre-configured packages from the Pentaho Customer Support Portal by using your Pentaho account login credentials. If you are unfamiliar with these details, consult the Welcome Kit provided to you by Pentaho customer support as part of enabling the Enterprise Edition of Pentaho Business Analytics.

The Pentaho Enterprise Console is included in the Pentaho Business Analytics package, so you do not need to download it separately.

The **Pentaho User Console plugins** items below refer to add-ons for the Pentaho User Console, such as Dashboard Designer, Interactive Reporting, Pentaho Mobile, and Pentaho Analyzer.

Windows

The packages you will need for a Windows server are:

- **BA Server (includes the Pentaho Enterprise Console):** biserver-ee-4.8.0-GA.zip
- **Pentaho Analyzer plugin:** paz-plugin-ee-4.8.0-GA.zip
- **Pentaho Dashboard Designer plugin:** pdd-plugin-ee-4.8.0-GA.zip
- **Pentaho Interactive Reporting plugin:** pir-plugin-ee-1.1.0-GA.zip
- **Pentaho Mobile:** pentaho-mobile-package-name.zip

The packages you will need for your Windows workstations are:

- **Report Designer:** prd-ee-3.9.0-GA.zip
- **Metadata Editor:** pme-ee-4.8.0-GA.zip
- **Schema Workbench:** psw-ee-3.4.1.zip
- **Design Studio for 32-bit Windows:** pds-ee-win-32-4.0.0-GA.zip
- **Design Studio for 64-bit Windows:** pds-ee-win-64-4.0.0-GA.zip
- **Design Studio plugins for existing Eclipse installations:** org.pentaho.designstudio.editors.actionsequence_4.0.0.GA.zip
- **Aggregation Designer:** /4.8.0-GA/client/windows/pad-ee-1.4.0-GA.zip
- **Data Integration:** pdi-ee-client-4.4.0-GA.zip

Supported Operating Systems Other Than Windows

The packages you will need for a Unix and Linux-based servers are:

- **BA Server (includes the Pentaho Enterprise Console):** biserver-ee-4.8.0-GA.tar.gz
- **Pentaho Analyzer plugin:** paz-plugin-ee-4.8.0-GA.tar.gz
- **Pentaho Dashboard Designer plugin:** pdd-plugin-ee-4.8.0-GA.tar.gz
- **Pentaho Interactive Reporting plugin:** pir-plugin-ee-1.1.0-GA.tar.gz
- **Pentaho Mobile:** pentaho-mobile-package-name.tar.gz

The packages you will need for your Linux or Solaris workstations are:

- **Report Designer:** prd-ee-3.9.0-GA.tar.gz
- **Metadata Editor:** pme-ee-4.8.0-GA.tar.gz
- **Schema Workbench:** psw-ee-3.4.1.tar.gz
- **Design Studio for 32-bit Linux:** pds-ee-linux-32-4.0.0-GA.tar.gz
- **Design Studio for 64-bit Linux:** pds-ee-linux-64-4.0.0-GA.tar.gz
- **Design Studio plugins for existing Eclipse installations:** org.pentaho.designstudio.editors.actionsequence_4.0.0.GA.zip
- **Aggregation Designer:** pad-ee-1.4.0-GA.tar.gz
- **Data Integration:** pdi-ee-client-4.4.0-GA.tar.gz

Creating a Pentaho System User on Linux

You may be able to use a different local user account, however you will have to modify all of the instructions to match that configuration.

Pentaho licenses are installed to an XML file in the home directory of the user that starts the Pentaho Enterprise Console. While you can use any system user to install licenses, it is easier to create a new user to start and stop the Pentaho Enterprise Console, and to install and update licenses with.

1. Open a local terminal on, or an OpenSSH session to the server you are hosting the BA Server on.

   ```
   ssh pgibbons@192.168.1.133
   ```

2. With root permissions, create a new user account called **pentaho**.

   Bash is not a requirement, but it is typically the shell that Linux users want to standardize on. On many Linux distributions, the default new user shell is /bin/sh or some equivalent (such as Dash) that may not use the ~/.bashrc configuration file that you will work with later. If you don't have or want to use Bash, adjust the instructions throughout this guide accordingly.

   ```
   sudo useradd -s /bin/bash -m pentaho
   ```

3. Set a password for the pentaho user (note that when using the **sudo** command, you must first supply the root password).

   ```
   sudo passwd pentaho
   ```

4. Verify that you can log in using the credentials you specified.

   ```
   su pentaho -
   ```

You now have a user account created specifically for running the BA Server and controlling Pentaho Enterprise Console start and stop scripts. You should stay logged into this new account to create the pentaho directory and perform all other installation tasks that do not explicitly require root access.

You must use this new user account for starting and stopping the Pentaho Enterprise Console, and for installing and updating licenses if you use the command line tool to manage them instead of the graphical interface in Pentaho Enterprise Console. If you create any RC or init scripts to start Pentaho Enterprise Console automatically at boot time, then you will have to write those scripts such that they start the service with the **pentaho** user credentials.

Local User Accounts on Linux

If you create a new local user account called **pentaho** as described in *Creating a Pentaho System User (Linux/Solaris)* , you must put the pentaho-solutions directory and store license information recorded by the Pentaho Enterprise Console in that directory. You must also adjust your `init` scripts to start your Web application server (which will run the BA Server) and the Pentaho Enterprise Console as this user. If you are unable to do this, you must modify the instructions in the rest of this guide to support your own custom user and directory configuration.

If you do not run the Pentaho Enterprise Console and the application server that runs the BA Server as the same system user, the BA Server will not be able to find the appropriate license information and will not operate with full functionality. If you need to launch the Pentaho Enterprise Console and the BA Server as separate users or services, you will have to set a **-D** parameter in your web application server's service configuration or startup scripts to specify a static location for your Pentaho license file. The Java parameter is **-Dpentaho.installed.licenses.file=/pentaho/.installedLicenses.xml**, though you will need to change this example to match your configuration.

Setting the PENTAHO_JAVA_HOME Variable on Linux

To ensure that Business Analytics will always use the correct Sun Java Runtime Environment, especially in software environments that contain multiple JREs, you must create a **PENTAHO_JAVA_HOME** system variable for your **pentaho** user account and point it to a supported JRE or JDK. If you do not set this variable, Business Analytics will attempt to use the JRE that the JAVA_HOME variable points to. We do not recommend that your BA Server installation rely on the **JAVA_HOME** variable, as this is the global default JRE for the entire system, and changing it could adversely affect other Java-based applications in the system. Therefore, Pentaho recommends that you set PENTAHO_JAVA_HOME as described in these steps.

1. Edit your **/etc/environment** file with a text editor.

   If you're using Solaris, you will have to set this environment variable through whatever means are available to you.
2. Add this line in a convenient place (replacing the path with the location of the JRE on your system): **export PENTAHO_JAVA_HOME=/usr/lib/jvm/java-6-sun**.
3. You must log out and log back into the operating system for the change to take effect.
4. Verify that the variable is properly set.

   ```
   env | grep PENTAHO_JAVA_HOME
   ```

Setting the PENTAHO_JAVA_HOME Variable on Windows

To ensure that Business Analytics will always use the correct Sun Java Runtime Environment, especially in software environments that contain multiple JREs, you must create a **PENTAHO_JAVA_HOME** system variable for your **pentaho** user account and point it to a supported JRE or JDK. If you do not set this variable, Business Analytics will attempt to use the JRE that the JAVA_HOME variable points to. We do not recommend that your BA Server installation rely on the **JAVA_HOME** variable, as this is the global default JRE for the entire system, and changing it could adversely affect other Java-based applications in the system. Therefore, Pentaho recommends that you set PENTAHO_JAVA_HOME as described in these steps.

1. In **Windows**, right-click on **Computer**, then select **Properties** from context menu, then click **Advanced System Settings**.

   The **System Properties** window will come up.
2. In the System Properties window, click the **Advanced** tab, then click **Environment Variables**.
3. In the System Variable section, click New.
4. A popup dialog will ask for a variable name and value. Type **PENTAHO_JAVA_HOME** into the name field.
5. In the value field, enter the directory for the JRE. An example of a JRE directory is `C:\Program Files\Java\jre6`. Click **OK**.
6. In the parent window, click **Apply Changes**.
7. You must restart your computer for the change to take effect.
8. Verify that the variable is properly set.

```
echo %PENTAHO_JAVA_HOME%
```

How to Check Your Java Version

Pentaho Business Analytics requires a Java Runtime Environment (JRE) or Java Development Kit (JDK). Follow this procedure to see which version of Java is installed on your system and configured to be the default Java executable. There may be multiple JREs or JDKs on your system, but only one can be set as the global default. If a particular JRE or JDK is not specified by an application on startup, the default is used. Pentaho establishes a specific system variable named **PENTAHO_JAVA_HOME** to declare which Java instance it will use.

1. Open a terminal or command prompt window.
2. Type this command in: **java -version** and press **Enter**.

   Along with the Java version, the bit-ness, 32-bit or 64-bit, and patch level also shows in the output. For example ...

```
java version "1.6.0_21"
Java(TM) SE Runtime Environment (build 1.6.0_21-b06)
Java HotSpot(TM) 64-Bit Server VM (build 17.0-b16, mixed mode)
```

Deploying the BA Server

Follow the directions below to install the BA Server and the Pentaho Enterprise Console onto your server. The examples assume a Linux or Solaris environment; adjust the paths accordingly for Windows-based machines.

Note: Solaris users: The default behavior of the tar utility in Solaris 10 is to truncate long file names when unpacking an archive created with GNU tar, as the Pentaho archives are. Therefore, you must use GNU tar instead of Solaris tar for this procedure. The path to GNU tar on Solaris 10 is typically `/usr/sfw/bin/gtar`. If you do not have gtar installed on your system, you will need to install it or some other GNU tar-compatible utility (such as star).

1. In a place that is accessible to the user account that will run and control the BA Server, create a `/pentaho/server/` directory.

   ```
   mkdir -p /home/pentaho/pentaho/server/
   ```

2. Change to the directory you just created.

   ```
   cd /home/pentaho/pentaho/server/
   ```

3. Untar or unzip the **biserver-ee-4.8-GA** archive from wherever you downloaded it to your current location.

   ```
   tar zxvf /home/amenethil/downloads/biserver-ee-4.8.0-GA.tar.gz
   ```

   This will create new **biserver-ee** and **enterprise-console** subdirectories.

The BA Server and Pentaho Enterprise Console are now extracted into the proper directories. Your servers are not completely configured yet. The next step is to configure the BA Server for your preferred solution repository database. To do that, you must add the appropriate database drivers to the BA Server and Enterprise Console, then create the system database, called the **solution repository** in your RDBMS. Follow the instructions for the supported databases that apply to your software environment.

Copying Solution Database JDBC Drivers

In order for your BA Server to connect to the system databases (referred to as the solution database, or solution repository) in the RDBMS of your choice, you need to add the RDBMS's JDBC driver library to the BA Server's and Enterprise Console's `lib` directory. The instructions in this section are focused solely on solution database configuration and connectivity with the BA Server.

Follow the below process to enable the BA Server to connect to an Oracle, MySQL, or PostgreSQL solution database.

1. Find or retrieve a JDBC driver JAR from your database vendor or third-party driver developer.

   Due to licensing restrictions, Pentaho does not distribute the necessary JDBC driver JARs. You can retrieve a JDBC driver from your database vendor. To that end, you may find these links helpful:

   - **Oracle:** *http://www.oracle.com/technetwork/topics/index.html*
   - **MySQL:** *http://www.mysql.com/downloads/connector/j/*
   - **PostgreSQL:** *http://jdbc.postgresql.org/download.html*

2. For the BA Server, copy the appropriate JDBC driver JAR file to the `/tomcat/lib/` directory for Tomcat, or the `/jboss/server/default/lib/` directory for JBoss.
3. For Enterprise Console, copy the driver JAR to the `/pentaho/server/enterprise-console/lib/` directory.

The BA Server and Enterprise Console have the necessary driver to communicate with your solution database.

Initializing a MySQL Database

In order to complete this process, you need to have the MySQL client installed on your machine, and the MySQL database server must be started. Ensure that you check your MySQL version for support compatibility. Refer to the *Compatibility Matrix: Supported Components* if you need supported version information.

Caution: You must use the **ASCII** character set in order for these scripts to execute properly. UTF-8 is known not to work because of text string length limitations.

This procedure creates and initializes Pentaho-related databases on your MySQL server. You must execute the scripts below in the order they are presented in the instructions, or else the solution database may not be properly initialized.

1. Navigate to the `/pentaho/server/biserver-ee/data/mysql5/` directory.
   There are several scripts in this directory. You should only run the ones mentioned below.
2. Execute the **create_quartz_mysql.sql** script using your preferred utility or process.

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

3. Execute the **create_repository_mysql.sql** script.

```
mysql -u root -p create_repository_mysql.sql
```

MySQL is now configured to act as a Pentaho BA Server solution repository.

MySQL Solution Repository Configuration

At this point you have installed the BA Server, and set up the RDBMS with the solution repository needed for the BA Server to run. The next steps configure the BA Server to use that repository. Follow these instructions to configure the BA Server to use MySQL as a solution repository with the default Pentaho security data access object. Even if you will end up using some other (security dao) authentication method, follow all of these directions so that you can verify that the BA Server works with a basic configuration.

1. Open the `/pentaho/server/biserver-ee/pentaho-solutions/system/applicationContext-spring-security-hibernate.properties` file in a text editor.
2. Edit the file and change the following values to those in this example, modifying the details for your specific MySQL configuration. Alternatively, you can copy over a MySQL-primed version of this file from /pentaho-solutions/system/dialects/mysql5/ and then change the details by hand.

```
jdbc.driver=com.mysql.jdbc.Driver
jdbc.url=jdbc:mysql://localhost:3306/hibernate
jdbc.username=hibuser
jdbc.password=password
hibernate.dialect=org.hibernate.dialect.MySQL5Dialect
```

3. Save and close the file, then edit `/pentaho/server/biserver-ee/tomcat/webapps/pentaho/META-INF/context.xml`
4. Change the hibernate and quartz **driverClassName** entries to `com.mysql.jdbc.Driver`
5. Change the hibernate **url** property to:

```
jdbc:mysql://localhost:3306/hibernate
```

Note: If you are using a different port number, change it accordingly in this and the next step. If your MySQL database server is not hosted on the same machine as your BA Server, you will need to change **localhost** to the hostname or IP address of your MySQL database server. There are other files you must modify as well; refer to the Changing the Solution Database Hostname later in this guide for further instruction.

6. Change the quartz **url** property to `jdbc:mysql://localhost:3306/quartz`
7. Replace the value of **validationQuery** with **"select 1 from dual"**.

```
validationQuery="select 1 from dual"
```

8. Save and close the file, then edit `/pentaho/server/biserver-ee/pentaho-solutions/system/hibernate/hibernate-settings.xml`
9. Replace the value of the <config-file> element with the following:

```
system/hibernate/mysql5.hibernate.cfg.xml
```

10. Save and close the file, then edit the database-specific configuration file that you just specified in the config-file element in the previous step.

11. Change the default hostname, port number, and user credentials to match your database configuration, then save and close the file.

    Even if you plan to use the default settings, open the file to verify that the defaults are what you expect them to be.

Configuring Quartz For MySQL

Follow this procedure to specify the appropriate JDBC jobStore delegate for MySQL.

1. Open the `/pentaho/server/biserver-ee/pentaho-solutions/system/quartz/quartz.properties` file in a text editor.
2. Find the following line in the properties file: `org.quartz.jobStore.driverDelegateClass =`. Replace the existing delegate portion with `org.quartz.impl.jdbcjobstore.StdJDBCDelegate`
3. Save the file and close the text editor.

The BA Server's scheduling (quartz) database is now configured to work with MySQL.

Configuring Audit Logging For MySQL

Audit logging is enabled by default in the BA Server. To configure it for MySQL, follow this process.

1. Copy all of the files from `/pentaho/server/biserver-ee/pentaho-solutions/admin/audit/dialects/mysql5/` to the `/pentaho/server/biserver-ee/pentaho-solutions/admin/audit/` directory, overwriting the existing files there.
2. Copy the **audit_sql.xml** file from `/pentaho/server/biserver-ee/pentaho-solutions/system/dialects/mysql5/` to `/pentaho/server/biserver-ee/pentaho-solutions/system/`, overwriting the existing version there.

Audit logging is configured for MySQL.

Initializing a PostgreSQL Database

In order to complete this process, you need to have the PostgreSQL native client tools installed, and the database server started. Your PostgreSQL configuration must also support logins from all users; this is not the default configuration in all cases, so you may have to edit your **pg_hba.conf** file to support this option.

This procedure will create and initialize Pentaho-related databases on your PostgreSQL server.

1. Navigate to the `/pentaho/server/biserver-ee/data/postgresql/` directory.

   There are several scripts in this directory. You should only run the ones mentioned below.
2. Edit the **create_repository_postgresql.sql** and **create_quartz_postgresql.sql** scripts and set the password references to the passwords you want to use for the Pentaho administrator and user accounts.

   The user account specified here is a generic account that the software uses to update the database when a user is logged in.
3. Execute the schedule database initialization script as the PostgreSQL root user by running this command from a terminal (change "postgres" to the name of your PostgreSQL root user): **psql -U postgres < create_quartz_postgresql.sql**.
4. Next, initialize your repository database: **psql -U postgres < create_repository_postgresql.sql**.

PostgreSQL is now configured to act as a Pentaho BA Server solution repository.

PostgreSQL Solution Repository Configuration

To this point you have installed the BA Server, and set up the RDBMS with the solution repository needed for the BA Server to run. The next steps configure the BA Server to use that repository. The default solution repository and scheduling database is PostgreSQL. Follow the below directions to configure the BA Server to use PostgreSQL as a solution repository with the default Pentaho security data access object. Even if you will end up using some other authentication method, follow all of these directions so that you can verify that the BA Server works with a basic configuration.

1. Open the `/pentaho/server/biserver-ee/pentaho-solutions/system/applicationContext-spring-security-hibernate.properties` file in a text editor.

2. Edit the file and change the following values to those in this example, modifying the details for your specific PostgreSQL configuration:

   Alternatively, you can copy over a PostegreSQL-primed version of this file from /pentaho-solutions/system/dialects/postgresql/ and then change the details by hand. This is the user and password that was modified in *Initializing a PostgreSQL Database* on page 16

   ```
   jdbc.driver=org.postgresql.Driver
   jdbc.url=jdbc:postgresql://localhost:5432/hibernate
   jdbc.username=hibuser
   jdbc.password=password
   hibernate.dialect=org.hibernate.dialect.PostgreSQLDialect
   ```

3. Save and close the file, then edit `/pentaho/server/biserver-ee/tomcat/webapps/pentaho/META-INF/context.xml`
4. Change the hibernate and quartz **driverClassName** entries to `org.postgresql.Driver`
5. Change the hibernate **url** property to:

   ```
   jdbc:postgresql://localhost:5432/hibernate
   ```

   

   Note: If you are using a different port number, change it accordingly in this and the next step. If your PostgreSQL database server is not hosted on the same machine as your BA Server, you will need to change **localhost** to the hostname or IP address of your PostgreSQL database server. There are other files you must modify as well; refer to the *Changing the Solution Database Hostname* on page 19 for further instruction.
6. Change the quartz **url** property to `jdbc:postgresql://localhost:5432/quartz`
7. Replace the value of **validationQuery** with **"select 1"**.

   ```
   validationQuery="select 1"
   ```

8. Save and close the file, then edit `/pentaho/server/biserver-ee/pentaho-solutions/system/hibernate/hibernate-settings.xml`
9. Replace the value of the <config-file> element with the following:

   ```
   system/hibernate/postgresql.hibernate.cfg.xml
   ```

10. Save and close the file, then edit the database-specific configuration file that you just specified in the config-file element.
11. Change the hibernate and quartz **driverClassName** entries to `org.postgresql.Driver`.
12. Change the hibernate **url** property to: `jdbc:postgresql://localhost:5432/hibernate`

    Even if you plan to use the default settings, open the file to verify that the defaults are what you expect them to be.

Configuring Quartz For PostgreSQL

Follow this procedure to replace the generic JDBC JobStore delegate with one specific to PostgreSQL.

1. Open the `/pentaho/server/biserver-ee/pentaho-solutions/system/quartz/quartz.properties` file in a text editor.
2. Find the following line in the properties file: `org.quartz.jobStore.driverDelegateClass =`
3. In the aforementioned line, replace the existing delegate with this: **org.quartz.impl.jdbcjobstore.PostgreSQLDelegate**
4. Save the file and close the text editor.

The BA Server's scheduling (quartz) database is now configured to work with PostgreSQL.

Configuring Audit Logging For PostgreSQL

Follow the below procedure to replace the generic JDBC driver for the Quartz JDBC JobStore.

1. Copy all of the files from `/pentaho/server/biserver-ee/pentaho-solutions/admin/audit/dialects/postgresql/` to the `/pentaho/server/biserver-ee/pentaho-solutions/admin/audit/` directory, overwriting the existing files there.

2. Copy the **audit_sql.xml** file from `/pentaho/server/biserver-ee/pentaho-solutions/system/dialects/postgresql/` to `/pentaho/server/biserver-ee/pentaho-solutions/system/`, overwriting the existing version there.

Audit logging is now configured for PostgreSQL.

Initializing an Oracle Database

In order to complete this process, Oracle should be installed and started, and you should have a basic familiarity with Oracle database management.

This procedure will initialize a Pentaho table space on your Oracle server.

1. Start the Oracle Database Configuration Manager and use it to create a database called **pentaho**.

   This database will house your BA Server solutions.
2. Navigate to the `/pentaho/server/biserver-ee/data/oracle10g/` directory.

   There are several scripts in this directory. You should only run the ones mentioned below.
3. Edit the **create_quartz_ora.sql** and **create_repository_ora.sql** scripts and replace the **admin/password** portion of the **conn** lines with the username and password you use to manage your Oracle database.

   The user credentials you type in here should have permission to create tables in the **pentaho** database, create users to access that database, and create and manage permissions for tables in it as well.
4. Edit the **datafile** path with the path to your Oracle installation, then save and close your text editor.
5. Run the SQL scripts with your preferred utility or process, in the order shown above.

   SQL*Plus is probably the easiest tool to use for this purpose.

Oracle is now configured to act as a Pentaho BA Server data repository.

Oracle Solution Repository Configuration

Follow the below directions to configure the BA Server to use Oracle as a solution repository with the default Pentaho security data access object. Even if you will end up using some other (security dao) authentication method, follow all of these directions anyway so that you can verify that the BA Server works with a basic configuration.

1. Open the `/pentaho/server/biserver-ee/pentaho-solutions/system/applicationContext-spring-security-hibernate.properties` file in a text editor.
2. Edit the file and change the following values to those in this example, modifying the details for your specific Oracle configuration:

```
jdbc.driver=oracle.jdbc.driver.OracleDriver
jdbc.url=jdbc:oracle:thin:@localhost:1521:XE
jdbc.username=hibuser
jdbc.password=password
hibernate.dialect=org.hibernate.dialect.Oracle10gDialect
```

3. Save and close the file, then edit `/pentaho/server/biserver-ee/pentaho-solutions/system/hibernate/hibernate-settings.xml`
4. Replace the value of the <config-file> element with the following:

```
system/hibernate/oracle10g.hibernate.cfg.xml
```

5. Save and close the file, then edit `/pentaho/server/biserver-ee/tomcat/webapps/pentaho/META-INF/context.xml`
6. Change the hibernate and quartz **driverClassName** entries to `oracle.jdbc.OracleDriver`
7. Change the hibernate **url** property to `jdbc:oracle:thin:@localhost:1521:XE`

   If you are using a different port number, change it accordingly in this and the next step. If your Oracle database server is not hosted on the same machine as your BA Server, you will need to change **localhost** to the hostname or IP address of your Oracle machine. Lastly, **XE** must change to match your Oracle service name. There are other files you must modify as well; refer to the *Changing the Solution Database Hostname* for further instruction.

8. Change the quartz **url** property to the following:

```
jdbc:oracle:thin:@localhost:1521:XE
```

9. Save and close the file, then edit the database-specific configuration file that you just specified in the config-file element.
10. Change the default hostname, port number, and user credentials to match your database configuration, then save and close the file.

    Even if you plan to use the default settings, open the file to verify that the defaults are what you expect.

Configuring Quartz For Oracle

Follow the below procedure to replace the generic JDBC driver for the Quartz JDBC JobStore.

1. Open the `/pentaho/server/biserver-ee/pentaho-solutions/system/quartz/quartz.properties` file in a text editor.
2. Find the following line in the properties file: `org.quartz.jobStore.driverDelegateClass = org.quartz.impl.jdbcjobstore.StdJDBCDelegate`
3. In the aforementioned line, replace the **org.quartz.impl.jdbcjobstore.StdJDBCDelegate** portion with this: **org.quartz.impl.jdbcjobstore.oracle.OracleDelegate**
4. Save the file and close the text editor.

The BA Server's scheduling (quartz) database is now configured to work with Oracle.

Configuring Audit Logging For Oracle

Audit logging is enabled by default in the BA Server. To configure it for Oracle, follow the below process.

1. Copy all of the files from `/pentaho/server/biserver-ee/pentaho-solutions/admin/audit/dialects/oracle10g/` to the `/pentaho/server/biserver-ee/pentaho-solutions/admin/audit/` directory, overwriting the existing files there.
2. Copy the audit_sql.xml file from `/pentaho/server/biserver-ee/pentaho-solutions/system/dialects/oracle10g/` to `/pentaho/server/biserver-ee/pentaho-solutions/system/`, overwriting the existing version there.

Audit logging is configured for Oracle.

Changing the Solution Database Hostname

The Pentaho BA Server's default configuration assumes that your solution database server and the Pentaho BA Server are going to be on the same machine, both accessible via **localhost**. If they are not on the same machine, you will have to modify some files in order to connect the two servers.

1. Open the `/pentaho/server/biserver-ee/tomcat/webapps/pentaho/META-INF/context.xml` file with a text editor.
2. Replace the **localhost** reference with your solution database server's hostname, domain name, or IP address.
3. Save and close the file, then edit the `/pentaho/server/biserver-ee/pentaho-solutions/system/applicationContext-spring-security-hibernate.properties` file and replace the **localhost** reference accordingly.
4. Save and close the file, then navigate to the `/pentaho/server/biserver-ee/pentaho-solutions/hibernate/` directory.
5. Edit the `database.hibernate.cfg.xml` file, where *database* is the name of your solution database (shown below) and change the **localhost** reference appropriately.

   The file you have to edit will be one of these:

   - mysql5.hibernate.cfg.xml
   - postgresql.hibernate.cfg.xml
   - oracle10g.hibernate.cfg.xml

The Pentaho BA Server is now configured to connect to a remote solution database.

Customizing Port Numbers

If you want to customize port numbers, follow the below procedure.

1. Navigate to the `/pentaho/server/biserver-ee/tomcat/conf/` directory
2. Edit the **server.xml** file found there, and search for **Define a non-SSL HTTP/1.1 Connector**. Change the port number in the **Connector port** element below that from **8080** to your preferred port number.

```
<!-- Define a non-SSL HTTP/1.1 Connector on port 8080 -->
    <Connector port="8080" maxHttpHeaderSize="8192"
               maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
               enableLookups="false" redirectPort="8443" acceptCount="100"
               connectionTimeout="20000" disableUploadTimeout="true" />
```

3. Save and close the file, then navigate to the `/pentaho/server/biserver-ee/tomcat/webapps/pentaho/WEB-INF/` directory and edit the **web.xml** file found there.
4. Change the **fully-qualified-server-url** entry to match the new port number you specified in server.xml.

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

5. Save and close the file, then navigate to the `/pentaho/server/enterprise-console/resource/config/` and edit the **console.properties** file found there.
6. Change the **console.start.port.number** and **console.stop.port.number** values to match the port numbers you want to run Enterprise Console on.

```
console.start.port.number=8088
console.stop.port.number=8033
```

Installing Pentaho User Console Plugins

In order to proceed with this task, you must have already purchased an Enterprise Edition license for the products you want to install. Skip this process if you do not have the requisite license.

The Pentaho User Console is built with a plugin architecture that enables you to expand its functionality with new client tools and functions. Follow the directions below to install any of the following plugins:

- Pentaho Analyzer
- Pentaho Dashboard Designer
- Pentaho Interactive Reporting
- Pentaho Mobile

Note: The Pentaho Analysis Enterprise Edition plugin is separate from Analyzer, and requires a different installation process. That process is found in *Installing the Analysis Enterprise Edition Plugin*.

1. If you have not already done so, download the plugin packages from the Pentaho Customer Support Portal. The installation materials list are defined in *Obtaining the Installation Matrials*.
2. Unpack the plugin packages to the `/pentaho/server/biserver-ee/pentaho-solutions/system/` directory.

   This will create a subdirectory for each plugin that you unpack.

The plugins you downloaded are now installed, though you will still need to merge your solution files and register your licenses.

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 `webkitgtik-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 webkitgtik-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.

Installing the Analysis Enterprise Edition Plugin

Follow the instructions below to install the Pentaho Analysis Enterprise Edition package, which contains Analysis engine enhancements for large ROLAP deployments. This procedure does not cover Pentaho Analyzer installation.

Note: If you are performing a manual WAR build and deployment, and you want to add the Pentaho Analysis Enterprise Edition JARs into the WAR pre-build, you can substitute `/biserver-manual-ee/build-resources/custom-pentaho-webapp/WEB-INF/` for `/WEB-INF/` references below.

1. If you have not already done so, retrieve the **pentaho-analysis-ee-4.8-GA** package from the Pentaho Customer Support Portal (login credentials were emailed to you with your Welcome Kit).
2. Unpack the archive to a temporary location.
3. If it is currently running, shut down the BA Server.

4. Copy only the following JARs from the `/pentaho-analysis-ee/lib/` directory to the `/tomcat/webapps/pentaho/WEB-INF/lib/` directory.
   - infinispan-core- 4.2.1.FINAL
   - jboss-transaction-api-1.0.1.GA
   - jcip-annotations-1.0
   - jgroups-2.12.0.CR5
   - marshalling-api-1.2.3.GA
   - memcached-0.0.1-PENTAHO
   - pentaho-analysis-ee-3.4.0-GA-obf
   - river-1.2.3.GA
5. Copy all of the configuration files from `/pentaho-analysis-ee/config/` to the `/tomcat/webapps/pentaho/WEB-INF/classes/ directory`.
6. Depending on the installation type; there would not be a **pentaho.war** (in archive based installations and executable based installations the **pentaho.war** is already deployed and the application will show as: `/tomcat/webapps/pentaho/`.
7. Remove the temporary **pentaho-analysis-ee** directory.

Pentaho Analysis Enterprise Edition is now installed with the default Infinispan configuration.

If you need to switch from Infinispan to Memcached, continue to the next section.

Switching to Memcached

In order to complete this procedure, you must have your own pre-configured Memcached instance. You should have also installed the Analysis Enterprise Edition package to your BA Server or standalone Mondrian engine.

If you already use the Memcached cache framework in your organization and would like to hook it up to the Pentaho Analysis ROLAP engine, follow the directions below to switch from the default Infinispan cache framework configuration.

Caution: Pentaho and Mondrian developers recommend against using Memcached. You are almost certain to have better performance with Infinispan.

1. If the BA Server or standalone Mondrian engine are running, shut them down now.
2. If you performed a default install of the Pentaho Analysis Enterprise Edition package, then you should have all of the required JARs installed to the BA or Mondrian server. If you aren't sure, verify now that the following JARs are present in the `/WEB-INF/lib/` directory inside of your deployed pentaho.war or Mondrian engine:
   - pentaho-analysis-ee
   - commons-lang
   - commons-io
   - commons-codec
   - pentaho-ee-dsc-core
   - memcached
3. Edit the **pentaho-analysis-config.xml** in the `/WEB-INF/classes/` directory inside the deployed pentaho.war or Mondrian engine, and change the value of **SEGMENT_CACHE_IMPL** to match the class name referenced below:

```
<entry key="SEGMENT_CACHE_IMPL">com.pentaho.analysis.segmentcache.impl.memcached.
MemcachedSegmentCache</entry>
```

4. Edit the **memcached-config.xml** in the `/WEB-INF/classes/` directory inside the deployed pentaho.war or Mondrian engine, and change the values of **SALT**, **SERVERS**, and **WEIGHT** to match your preference:

```
                    <entry key="SALT">YOUR SECRET SALT VALUE HERE</entry>
                    <entry key="SERVERS">192.168.0.1:1642,192.168.0.2:1642</entry>
                    <entry key="WEIGHTS">1,1</entry>
```

Your Pentaho Analysis Enterprise Edition instance is now configured to use Memcached for ROLAP segment caching.

Adding Linux and OS X Support for Printing Analyzer Geo Maps

There is an experimental function in Analyzer that enables you to print Geo Map visualizations. Follow the directions below to enable support on Linux and OS X. Windows configuration is covered in *Adding Windows Support for Printing Analyzer Geo Maps* on page 23.

Note: Pentaho requires a specific version of **wkhtmltoimage** in order for this print function to work correctly. Even if you have wkhtmltoimage installed on your machine, install the Pentaho-supplied version as explained below. It will be explicitly called by Analyzer, and will not interfere with your existing wkhtmltoimage instance.

Note: If you used the Business Analytics graphical installer, **skip all steps except the last two.**

1. Navigate to the **wkhtmltoimage** directory in your Pentaho installation. For archive installations, this is `/pentaho/biserver-ee/third-party-tools/wkhtmltoimage/`, and for manual deployments it is `/biserver-manual-ee/build-resources/third-party-tools/wkhtmltoimage/`.
2. For OS X, unpack the **wkhtmltoimage_mac.tar.bz2** package; for 32-bit Linux, unpack the **wkhtmltoimage_linux_i386.tar.bz2** package; for 64-bit Linux, unpack the **wkhtmltoimage_linux_amd64.tar.bz2** package.
3. There is only one file in this archive: the executable **wkhtmltoimage**. If the executable has an extension, such as **i386** or **amd64**, rename the file to **wkhtmltoimage**.
4. Move the executable to: `/home/pentaho/pentaho/server/biserver-ee/wkhtmltoimage`. directory.
5. Edit the `/pentaho-solutions/system/pentaho.xml` file and set the value of **<html-to-image-tool>** to the full path to the **wkhtmltoimage** executable that you just unpacked.

   ```
   <html-to-image-tool>/home/pentaho/pentaho/server/biserver-ee/third-party-tools/
   wkhtmltoimage/wkhtmltoimage</html-to-image-tool>
   ```

6. Save and close the file, then edit the `/pentaho-solutions/system/analyzer/settings.xml` file and uncomment the following line:

   ```
   <!--open_layers>geomapexport</open_layers-->
   ```

7. Save and close the settings.xml file.

You have enabled the experimental print feature for Analyzer Geo Map visualizations.

Adding Windows Support for Printing Analyzer Geo Maps

There is an experimental function in Analyzer that enables you to print Geo Map visualizations. Follow the below directions to enable support on Windows. Linux and OS X configuration are covered in *Adding Linux and OS X Support for Printing Analyzer Geo Maps* on page 23.

Note: Pentaho requires a specific version of **wkhtmltoimage** in order for this print function to work correctly. Even if you have wkhtmltoimage installed on your machine, install the Pentaho-supplied version as explained below. It will be explicitly called by Analyzer, and will not interfere with your existing wkhtmltoimage instance.

Note: If you used the Business Analytics graphical installer, **skip all steps except the last two.**

1. Navigate to the **wkhtmltoimage** directory in your Pentaho installation. For archive installations, this is `\pentaho\biserver-ee\third-party-tools\wkhtmltoimage\`, and for manual deployments it is `\biserver-manual-ee\build-resources\third-party-tools\wkhtmltoimage\`.
2. Run the **wkhtmltoimage_windows_installer.exe** executable.
3. Accept the license agreement to continue.
4. Uncheck the **wkhtmltopdf** option, then click **Next**.
5. For the **Destination folder**, type in or navigate to the following path, creating the directories if they do not exist: `C:\pentaho\biserver-ee\third-party-tools\wkhtmltoimage\`. Once you have navigated to the path, click **Install**.

6. When the installation is complete, click **Close**, then edit the `\pentaho-solutions\system\pentaho.xml` file and set the value of **<html-to-image-tool>** to the full path to the **wkhtmltoimage** executable that you specified in the installer, including the executable name of the package, which is **wkhtmltoimage**.

```
<html-to-image-tool>C:\pentaho\biserver-ee\third-party-tools\wkhtmltoimage
\wkhtmltoimage</html-to-image-tool>
```

7. Close that file, then edit the `\pentaho-solutions\system\analyzer\settings.xml` file and uncomment the following line:

```
<!--open_layers>geomapexport</open_layers-->
```

8. Save and close the settings.xml file.

You have enabled the experimental print feature for Analyzer Geo Map visualizations.

Initial Startup of the BA Server and Pentaho Enterprise Console

The basic installation procedures are now complete, so it's time to start the BA Server and Pentaho Enterprise Console so that you can proceed with low-level system configuration.

1. Execute the `/pentaho/server/biserver-ee/start-pentaho` script.

   This script has a **.bat** extension on Windows, and a **.sh** extension on Linux, Solaris, and OS X.
2. Execute the `/pentaho/server/enterprise-console/start-pec` script.

   This script has a **.bat** extension on Windows, and a **.sh** extension on Linux, Solaris, and OS X. It may take a few moments to compete the startup sequence.

Both the Pentaho BA Server and the Pentaho Enterprise Console server are now running and should be accessible via a Web browser.

Post-Install Configuration

After you've successfully installed the BA Server, you must set up your configuration information and other details. All of this is done through the Pentaho Enterprise Console.

Note: In a manual deployment scenario, all paths in the Enterprise Console configuration screen -- Web-App, Solution Directory, Backup Directory -- must be absolute (full paths). Relative paths will not validate.

1. Open a Web browser and navigate to **http://localhost:8088** (change **localhost** to the IP address, hostname, or domain name of your BA Server).
2. Log in as **admin**, using **password** as the password.
3. Click the **+** (plus) button in the upper right corner of the Subscriptions section.

   An **Install License** dialogue will appear.
4. Click **Browse**, then navigate to the location you saved your LIC files to, click on one of the LIC files, then click **Open**.

   LIC files for each of your supported Pentaho products were emailed to you along with your Pentaho Welcome Kit. If you did not receive this email, or if you have lost these files, contact your Pentaho support representative. If you do not yet have a support representative, contact the Pentaho salesperson you were working with.
5. Click **OK**.

   The Setup screen will change according to the LIC file you installed. **Repeat the LIC file installation process for each license you have purchased.**

   

   Note: After installing license files, Linux and Solaris users must restart the application server before they can use any functions in the Pentaho User Console. The Pentaho Enterprise Console does not have to be restarted, so you can continue configuring the system, but you must restart JBoss or Tomcat when you are finished.
6. In the **Solution Directory** field, enter `home/pentaho/server/biserver-ee/pentaho-solutions/`, or whatever the path to your pentaho-solutions directory is.
7. In the **Backup Directory** field, type in the location that you'd like to save Pentaho Enterprise Console backup data to.

This can be any local directory that your application server has permissions to write to. You cannot, however, use relative paths in this or any other configuration field in this screen -- **all paths must be absolute**.

8. In the **Pentaho Web-App Path** field, enter `/tomcat/webapps/pentaho/`, or whatever the path to the unpacked pentaho.war directory is for your application server.
9. In the **Platform Administrator User Name** field, type in **admin**, or if you already have Pentaho User Console accounts established, type in the account name that you will use to manage reports and schedules.

   This user account does not have to exist yet, and you can change this value later.
10. Click **OK**.

    The rest of the settings in this screen do not need to change right now.
11. Click **Configuration** in the menu on the left side of Pentaho Enterprise Console.
12. Click the **Web Settings** tab at the top of the screen.
13. Change the **Fully Qualified Server URL** setting to match your server's hostname, domain name, or IP address. Do not change the directory or port number.

    If you need to change the port number, consult the *Pentaho Business Analytics Administrator's Guide*.

Your Pentaho BA Server has a minimal configuration. If you need to return to this screen, click on the wrench/screwdriver icon in the upper right corner of the screen.

Your administration work is only beginning; you should now consult the *Pentaho Business Analytics Administrator's Guide* to learn your way around the Pentaho Enterprise Console.

Preparing a Headless Linux or Solaris Server

There are two headless server scenarios that require special procedures on Linux and Solaris systems. One is for a system that has no video card; the other is for a system that has a video card, but does not have an X server installed. In some situations -- particularly if your server doesn't have a video card -- you will have to perform both procedures in order to properly generate reports with the BA Server.

Systems without video cards

The **java.awt.headless** option enables systems without video output and/or human input hardware to execute operations that require them. To set this application server option when the BA Server starts, open the `/pentaho/server/biserver-ee/start-pentaho.sh` script with a text editor, then add the following item to the list of CATALINA_OPTS parameters: **-Djava.awt.headless=true**.

The entire line should look something like this:

```
export CATALINA_OPTS="-Djava.awt.headless=true -Xms256m -Xmx768m -XX:MaxPermSize=256m
 -Dsun.rmi.dgc.client.gcInterval=3600000 -Dsun.rmi.dgc.server.gcInterval=3600000"
```

Note: If you do not have an X server installed, you must also follow the below instructions.

Systems without X11

In order to generate charts, the Pentaho Reporting engine requires functionality found in X11. If you are unwilling or unable to install an X server, you can install the **xvfb** package instead. The **xvfb** package provides X11 framebuffer emulation, which performs all graphical operations in memory instead of sending them to the screen.

Use your operating system's package manager to properly install xvfb.

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.

Post-Install Configuration

After you've successfully installed the BA Server, you must set up your configuration information and other details. All of this is done through the Pentaho Enterprise Console.

Note: In a manual deployment scenario, all paths in the Enterprise Console configuration screen -- Web-App, Solution Directory, Backup Directory -- must be absolute (full paths). Relative paths will not validate.

1. Open a Web browser and navigate to **http://localhost:8088** (change **localhost** to the IP address, hostname, or domain name of your BA Server).
2. Log in as **admin**, using **password** as the password.
3. Click the **+** (plus) button in the upper right corner of the Subscriptions section.

   An **Install License** dialogue will appear.
4. Click **Browse**, then navigate to the location you saved your LIC files to, click on one of the LIC files, then click **Open**.

   LIC files for each of your supported Pentaho products were emailed to you along with your Pentaho Welcome Kit. If you did not receive this email, or if you have lost these files, contact your Pentaho support representative. If you do not yet have a support representative, contact the Pentaho salesperson you were working with.
5. Click **OK**.

   The Setup screen will change according to the LIC file you installed. **Repeat the LIC file installation process for each license you have purchased.**

   

   Note: After installing license files, Linux and Solaris users must restart the application server before they can use any functions in the Pentaho User Console. The Pentaho Enterprise Console does not have to be restarted, so you can continue configuring the system, but you must restart JBoss or Tomcat when you are finished.
6. In the **Solution Directory** field, enter `home/pentaho/server/biserver-ee/pentaho-solutions/`, or whatever the path to your pentaho-solutions directory is.
7. In the **Backup Directory** field, type in the location that you'd like to save Pentaho Enterprise Console backup data to.

   This can be any local directory that your application server has permissions to write to. You cannot, however, use relative paths in this or any other configuration field in this screen -- **all paths must be absolute**.
8. In the **Pentaho Web-App Path** field, enter `/tomcat/webapps/pentaho/`, or whatever the path to the unpacked pentaho.war directory is for your application server.
9. In the **Platform Administrator User Name** field, type in **admin**, or if you already have Pentaho User Console accounts established, type in the account name that you will use to manage reports and schedules.

   This user account does not have to exist yet, and you can change this value later.
10. Click **OK**.

    The rest of the settings in this screen do not need to change right now.
11. Click **Configuration** in the menu on the left side of Pentaho Enterprise Console.
12. Click the **Web Settings** tab at the top of the screen.
13. Change the **Fully Qualified Server URL** setting to match your server's hostname, domain name, or IP address. Do not change the directory or port number.

    If you need to change the port number, consult the *Pentaho Business Analytics Administrator's Guide*.

Your Pentaho BA Server has a minimal configuration. If you need to return to this screen, click on the wrench/screwdriver icon in the upper right corner of the screen.

Your administration work is only beginning; you should now consult the *Pentaho Business Analytics Administrator's Guide* to learn your way around the Pentaho Enterprise Console.

Installing the Pentaho Client Tools

Follow the directions below to install the Pentaho client tools onto your user workstations. The examples assume a Linux or Solaris environment; adjust the paths accordingly for Windows-based machines.

1. In a place that is accessible to the user accounts that will run the client tools, create a `/pentaho/design-tools/` directory.

   ```
   mkdir -p /home/amenethil/pentaho/design-tools/
   ```

2. Change to the directory you just created.

   ```
   cd /home/amenethil/pentaho/design-tools/
   ```

3. Untar or unzip the archive packages for all of the client tools you have support entitlements for.

   ```
   tar zxvf /home/amenethil/downloads/prd-ee-3.9.0-GA.tar.gz
   ```

There are no further required steps for installing the Pentaho client tools, but you may want to create desktop shortcuts to them for your users.

Installing the Pentaho Design Studio Eclipse Plugin

You must have the Java Runtime Environment (or Development Kit) version 1.6, and Eclipse version 3.2 or later installed before continuing.

Pentaho Design Studio is typically installed as a complete Eclipse-based application, including the Eclipse platform. However, if you already have your own Eclipse instance, you can install the Design Studio plugin for Eclipse instead.

1. Exit Eclipse if it is running.
2. If you have not done so already, download the **org.pentaho.designstudio.editors.actionsequence4.8.0.GA.zip** package from the Pentaho Customer Support Portal.
3. Unzip both files into your Eclipse home directory.
4. Verify that the plugin is present by looking for a directory that has a name beginning with **org.pentaho.designstudio.editors** in the plugins directory of your Eclipse home directory.
5. Start Eclipse.

Design Studio is now installed as an Eclipse plugin.

Preparing for Production

The default condition of the BA Server is designed to support quick and easy evaluation by new customers. Once you move from evaluation to production, you may want to remove some of the evaluation-specific features in the BA Server. The instructions in this section explain how to remove them.

Removing Pentaho Sample Data and Solutions

By default, Pentaho provides a sample data source and a solution directory filled with example content. These samples are provided for evaluation and testing. Once you are ready to move from an evaluation or testing scenario to development or production, you can remove the sample content. Follow the instructions below to completely remove the Pentaho sample data and solutions.

1. Stop the BA Server.
2. If they exist, delete the `/pentaho/server/biserver-ee/pentaho-solutions/steel-wheels/` and `/pentaho/server/biserver-ee/pentaho-solutions/bi-developers/` directories.

   These directories contain sample content for CDF and BA Server plugins. They are not present in typical manual deployment scenarios.
3. Edit the `/pentaho/WEB-INF/web.xml` file inside of the deployed pentaho.war.

   As laid down by the Pentaho graphical installer and archive packages, this path should be `/pentaho/server/biserver-ee/tomcat/webapps/pentaho/WEB-INF/web.xml`. If you performed a manual WAR build and deployment, then you must adjust the path to fit your configuration.
4. Remove the sample servlets by commenting them out or removing them from the file:

```
<!-- [BEGIN SAMPLE SERVLETS] -->
 <servlet>
  <servlet-name>UpdateSampleEmails</servlet-name>
  <jsp-file>/jsp/UpdateSampleEmails.jsp</jsp-file>
 </servlet>

 <servlet>
  <servlet-name>Widgets</servlet-name>
  <jsp-file>/jsp/Widgets.jsp</jsp-file>
 </servlet>

 <servlet>
  <servlet-name>SampleDashboard</servlet-name>
  <jsp-file>/jsp/SampleDashboard.jsp</jsp-file>
 </servlet>

 <servlet>
  <servlet-name>ChartSamplesDashboard</servlet-name>
  <jsp-file>/jsp/ChartSamplesDashboard.jsp</jsp-file>
 </servlet>

 <servlet>
  <servlet-name>SampleDrill</servlet-name>
  <jsp-file>/jsp/SampleDrill.jsp</jsp-file>
 </servlet>

 <servlet>
  <servlet-name>SWDashboard</servlet-name>
  <jsp-file>/jsp/SWDashboard.jsp</jsp-file>
 </servlet>

 <servlet>
  <servlet-name>Map</servlet-name>
  <jsp-file>/jsp/Map.jsp</jsp-file>
 </servlet>

 <servlet>
  <servlet-name>SampleFlashDashboard</servlet-name>
  <jsp-file>/jsp/SampleFlashDashboard.jsp</jsp-file>
```

```
 </servlet>
<!-- [END SAMPLE SERVLETS] -->
```

5. Remove the sample servlet mappings in the same fashion:

```
<!-- [BEGIN SAMPLE SERVLET MAPPINGS] -->
 <servlet-mapping>
  <servlet-name>UpdateSampleEmails</servlet-name>
  <url-pattern>/UpdateSampleEmails</url-pattern>
 </servlet-mapping>

 <servlet-mapping>
  <servlet-name>Widgets</servlet-name>
  <url-pattern>/Widgets</url-pattern>
 </servlet-mapping>

 <servlet-mapping>
  <servlet-name>SampleDashboard</servlet-name>
  <url-pattern>/SampleDashboard</url-pattern>
 </servlet-mapping>

 <servlet-mapping>
  <servlet-name>ChartSamplesDashboard</servlet-name>
  <url-pattern>/ChartSamplesDashboard</url-pattern>
 </servlet-mapping>

 <servlet-mapping>
  <servlet-name>SampleDrill</servlet-name>
  <url-pattern>/SampleDrill</url-pattern>
 </servlet-mapping>

 <servlet-mapping>
  <servlet-name>SWDashboard</servlet-name>
  <url-pattern>/SWDashboard</url-pattern>
 </servlet-mapping>

 <servlet-mapping>
  <servlet-name>Map</servlet-name>
  <url-pattern>/Map</url-pattern>
 </servlet-mapping>

 <servlet-mapping>
  <servlet-name>SampleFlashDashboard</servlet-name>
  <url-pattern>/SampleFlashDashboard</url-pattern>
 </servlet-mapping>
<!-- [END SAMPLE SERVLET MAPPINGS] -->
```

6. Remove the **SampleDataStartupListener**:

```
<listener>
    <listener-class>org.pentaho.platform.web.http.context.HsqldbStartupListener</
listener-class>
</listener>
```

7. Remove the **hsqldb-databases** section:

```
<!-- [BEGIN HSQLDB DATABASES] -->
    <context-param>
        <param-name>hsqldb-databases</param-name>
        <param-value>sampledata@../../data/hsqldb/sampledata</param-value>
    </context-param>
<!-- [END HSQLDB DATABASES] -->
```

8. Remove the **SystemStatusFilter** filter:

Note: This is not part of the Pentaho samples; it provides error status messages that are only useful for development and testing purposes, and should be removed from a production system.

```
<filter>
```

```
    <filter-name>SystemStatusFilter</filter-name>
    <filter-class>com.pentaho.ui.servlet.SystemStatusFilter</filter-class>
    <init-param>
        <param-name>initFailurePage</param-name>
        <param-value>InitFailure</param-value>
        <description>This page is displayed if the PentahoSystem fails to properly
 initialize.</description>
    </init-param>
</filter>
```

9. Remove the filter mapping for the **SystemStatusFilter**:

```
<filter-mapping>
    <filter-name>SystemStatusFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

10. Save and close the web.xml file.
11. Delete the `/pentaho/server/biserver-ee/data/` directory.

    This directory contains a sample database, control scripts for that database and the environment settings it needs to run, and SQL scripts to initialize a new Pentaho solution repository.

Your BA Server instance is now cleaned of samples and development/testing pieces, and is streamlined for production.

Continue to *Migrating Your Solutions* to finish the upgrade process.

Switching to a Production Login Screen

The default Pentaho User Console login screen contains information and instructions for evaluators. Pentaho allows this extra information to be hidden via configuration; follow the directions below to hide the extra information. If you have chosen to perform a manual deployment, this information is hidden by default.

1. Shut down the BA Server and Enterprise Console if they are currently running.
2. Open a terminal or file browser window and navigate to the `/pentaho-solutions/system/` directory.

   If you installed via the archive package or the graphical installer, the full path is `/pentaho/server/biserver-ee/pentaho-solutions/system/`.
3. Edit the **pentaho.xml** file.
4. Find the **login-show-sample-users-hint** node and change its value to **false**.
5. Start the BA Server and Pentaho Enterprise Console.

You now have a login screen that has been scrubbed of evaluation content.

Compatibility Matrix: Supported Components

Pentaho aims to accommodate our clients' diverse computing environments. This list provides details about the environment components and versions we support. If you have questions about your particular computing environment, please contact Pentaho support.

Client

Pentaho client software is hardware-independent and runs on client-class computers that comply with these specifications for minimum hardware and required operation systems.

Pentaho Software
Pentaho Aggregation Designer
Pentaho Data Integration
Pentaho Design Studio
Pentaho Metadata Editor
Pentaho Report Designer
Pentaho Schema Workbench

Hardware—32 or 64 bit	Operating System—32 or 64 bit
Processors: • Apple Macintosh Dual-Core • Intel EM64T or AMD64 Dual-Core **RAM:** 2 GB RAM **Disk Space:** 2 GB free after installation	• Apple Macintosh OS 10.7 & 10.8 • Microsoft Windows 7 • Ubuntu Server 10.X and 12.X

Server

Pentaho server software is hardware-independent and runs on server-class computers that comply with these specifications for minimum hardware and required operation systems.

Pentaho Software
Pentaho Business Analysis Server
Pentaho Data Integration Server
Pentaho Enterprise Console

Hardware—64 bit	Operating System—64 bit
• Apple Macintosh Pro Quad-Core or Macintosh Mini Quad-Core • Intel EM64T or AMD64 Dual-Core **RAM:** 8 GB with 4 GB dedicated to Pentaho servers, 1 GB to Pentaho Enterprise Console **Disk Space:** 20 GB free after installation	• Apple Macintosh OS X Server 10.6 & 10.7 • CentOS Linux 5 & 6 • Microsoft Windows 2008 Server R1 & R2 • Red Hat Enterprise Linux 5 & 6 • Solaris 10 • Ubuntu Server 10.X & 12.X

Embedded Software

When embedding Pentaho software into other applications, the computing environment should comply with these specifications for minimum hardware and required operation systems.

Pentaho Software
Embedded Pentaho Reporting
Embedded Pentaho Analysis
Embedded Pentaho Data Integration

Hardware—32 or 64 bit	Operating System—32 or 64 bit
Processors: • Apple Macintosh Pro Quad-Core or Macintosh Mini Quad-Core • Intel EM64T or AMD64 Dual-Core **RAM:** 8 GB with 4 GB dedicated to Pentaho servers **Disk Space:** 20 GB free after installation	• Apple Macintosh OS X Server 10.6 & 10.7 • CentOS Linux 5 & 6 • Microsoft Windows 2008 Server R1 & R2 • Microsoft Windows 7 • Red Hat Enterprise Linux 5 & 6 • Solaris 10 • Ubuntu Server 10.X & 12.X

Application Servers

Servers to which you deploy Pentaho software must run one of these application servers.

Pentaho Software	Application Server
Pentaho Business Analysis Server	• Jboss 5.1.x • Tomcat 6.0.x
Pentaho Data Integration Server	Tomcat 6.0.x

Solution Database Repositories

Pentaho software stores processing artifacts in these solution database repositories.

Pentaho Software	Database Repository
Pentaho Business Analysis Server	• MySQL 5.x • Oracle 10g/11i • PostgreSQL 8.x & 9.1.x*
Pentaho Data Integration Server	Integrated Pentaho-specific, H2 1.2.131

*Default installed solution database

Data Sources

Pentaho software connects to these relational and non-relational data sources.

Pentaho Software	Data Source
Pentaho Reporting	• JDBC 3** • ODBC • OLAP4J • XML • Pentaho Analysis • Pentaho Data Integration • Pentaho Metadata
Pentaho Business Analysis Server, Action Sequences	• Relational (JDBC) • Hibernate • Javascript

Pentaho Software	Data Source
	• Metadata (MQL) • Mondrian (MDX) • XML (XQuery) • Security User/Role List Provider • Data Integration Steps (PDI) • Other Action Sequences • Web Services • XMLA

Pentaho Data Integration	• JDBC 3**
	• OLAP4J
	• Salesforce
	• XML
	• CSV
	• Microsoft Excel
	• Pentaho Analysis
	• Apache Hadoop 0.20.2 & 0.20.203.0***
	• Cloudera CDH3u4***
	• CDH4***
	• MapR 1.1.3 & 1.2.0
	• Cassandra distributions
	  • Apache 1.1.2
	  • DataStax 1.1.2
	• MongoDB 2.0.4

**Use a JDBC 3.x compliant driver that is compatible with SQL-92 standards when communicating with relational data sources. For your convenience, we provide a list of drivers used to get data from relational JDBC databases.

***From one of these distributions: HBase 0.90.5 and Hive 0.7.1

SQL Dialect-Specific

Pentaho software generates dialect-specific SQL when communicating with these data sources.

Pentaho Software	Data Source
Pentaho Analysis	• Access • DB2 • Derby • Firebird • Greenplum • Hive • Hsqldb • Infobright • Informix • Ingres • Interbase • LucidDb • MicrosoftSqlServer • MySql • Neoview • Netezza • Oracle • PostgreSQL

Pentaho Software	Data Source
	• SqlStream • Sybase • Teradata • Vectorwise • Vertica • Other SQL-92 compliant****
Pentaho Metadata	• DB2 • Firebird • H2 • Hypersonic • Ingres • MS Access • ASSQL • MSSQLNative • MySQL • Netezza • Oracle • PostgresSQL • Sybase • Other SQL-92 compliant****
Pentaho Data Integration	• Apache Derby • AS/400 • InfiniDB • Exasol 4 • Firebird SQL • Greenplum • H2 • Hypersonic • IBM DB2 • Infobright • Informix • Ingres • Ingres VectorWise • LucidDB • MaxDB (SAP DB) • MonetDB • MySQL • MS SQL Server • Neoview • Netezza • Oracle • Oracle RDB • PostgreSQL • SQLite • Teradata • UniVerse database • Vertica • Other SQL-92 compliant****

****If your data source is not in this list and is compatible with SQL-92, Pentaho software uses a generic SQL dialect.

Security

Pentaho software integrates with these third-party security authentication systems.

Pentaho Software	Authentication System
Pentaho Business Analysis Server Pentaho Enterprise Console	• Active Directory • CAS • Integrated Microsoft Windows Authentication • LDAP • RDBMS
Pentaho Data Integration Server	• Active Directory • LDAP • RDBMS

Java Virtual Machine

All Pentaho software, except the Pentaho Mobile App, requires the Sun/Oracle version 1.6 (6.0) distribution of the Java Runtime Environment (JRE) or Java Development Kit (JDK).

Web Browsers

Pentaho supports these major versions of Web browsers that are publicly available six weeks prior to when Pentaho begins to finalize a release. We also support the preceding major version.

Pentaho Software	Web Browser
Pentaho User Console Pentaho Enterprise Console Pentaho Report Designer*****	• Apple Safari 5.x • Google Chrome 19 • Microsoft Internet Explorer 8 & 9 • Mozilla Firefox 13 & 14

*****Requires a web browser to preview the exported HTML reports.

Mobile Apps

Pentaho mobile apps run on the Apple iPad 2 and 3 using iOS 5.x and 6.

JDBC Drivers

JDBC Drivers

This reference is a continuous work in progress. If you are viewing it in the Pentaho InfoCenter and see something that is not correct, know of a driver that is not listed here, or have a tip you want to share, please let us know by using the comments fields found in the bottom right corner.

Database	Vendor	URL
Apache Derby	IBM	*http://db.apache.org/derby/derby_downloads.html*
Cache'	InterSystems	*http://www.cachemonitor.de/intersystems-documentation/cache-jdbc-driver*
CUBRID	CUBRID	*http://www.cubrid.org/?mid=downloadsitem=jdbc_driver*
Daffodil DB	Daffodil Software	*http://sourceforge.net/projects/daffodildb/*

Database	Vendor	URL
DB2 AS/400	IBM	*http://www-03.ibm.com/systems/i/software/toolbox/*
DB2 Universal Database	IBM	*http://www-306.ibm.com/software/data/db2/java*
Firebird	Firebird Foundation	*http://www.firebirdsql.org/en/jdbc-driver/*
FrontBase	FrontBase	*http://www.frontbase.com/cgi-bin/WebObjects/FBWebSite*
Greenplum	EMC2	*http://jdbc.postgresql.org/download.html*
H2 Database	H2	*http://www.h2database.com*
Hive	Apache	*http://hive.apache.org/*
HSQLDB	HyperSQL	*http://sourceforge.net/projects/hsqldb/*
Informix	IBM	*http://www-01.ibm.com/software/*
Ingres	Actian	*http://esd.actian.com/product/drivers/JDBC/java*
InterBase	Embarcadero	*http://edn.embarcadero.com*
jTDS Free MS Sybase	jTDS	*http://jtds.sourceforge.net/*
LucidDB	DynamoDB	*http://www.dynamobi.com/c/downloads/stable/*
MaxDB	SAP	*http://maxdb.sap.com*
Mckoi	Mckoi SQL Database	*http://www.mckoi.com/originalmckoisql/index.html*
Mimer	Mimer Information Technology	*http://www.mimer.com*
MonetDB	MonetDB	*http://www.monetdb.org/*
MySQL	Oracle	*http://dev.mysql.com/downloads/connector/j/*
Neoview	HP	*https://h20392.www2.hp.com/portal/swdepot/displayProductInfo.do?productNumber=NEO10*
Netezza	IBM	*http://www.netezza.com*
OpenBase SQL	OpenBase International	*http://www.openbase.com/index.php/products/downloads*
Oracle	Oracle	*http://www.oracle.com/technetwork/database/features/jdbc/index.html*
Pervasive	Pervasive	*http://www.pervasivedb.com/download/Pages/PDBDownloads.aspx*
PostgreSQL	PostgreSQL Global Development Group	*http://jdbc.postgresql.org/*
SAP DB	SAP DB	*http://www.sapdb.org/sap_db_jdbc.htm*
SQLite	Xerial	*http://www.xerial.org/trac/Xerial/wiki/SQLiteJDBC*

Database	Vendor	URL
SQL Server	Microsoft	*http://msdn.microsoft.com/en-us/sqlserver/aa937724.aspx*
Sybase ASE	SAP	*http://www.sybase.com/products/allproductsa-z/softwaredeveloperkit/jconnect*
Sybase SQL Anywhere	SAP	*http://www.sybase.com/products/allproductsa-z/softwaredeveloperkit/jconnect*
SmallSQL	SmallSQL	*http://www.smallsql.de/download.html*
Teradata	Teradata	*http://downloads.teradata.com/download/connectivity/jdbc-driver*
Vertica	HP	*http://www.vertica.com*

Apache Derby

Vendor Name	Details	
Recommended Native Driver		
IBM	**Company URL** *http://www.ibm.com*	
	Driver URL *http://db.apache.org/derby/derby_downloads.html*	
	JDBC URL Syntax by Type Server—jdbc:derby://<server>[:<port>]/<databaseName>[;<URL attribute>=>value>] Embedded—jdbc:derby:<databaseName>[;create=true]	**Default Port** 1527
	JDBC Class org.apache.derby.jdbc.ClientDriver org.apache.derby.jdbc.EmbeddedDriver	**JDBC JAR File Name** derby.jar
	Shipped with Pentaho Products Pentaho Data Integration	
	Comments Open source database	

Caché

Vendor Name	Details
Recommended Native Driver	
InterSystems	**Company URL** *http://www.cachemonitor.de*
	Driver URL *http://www.cachemonitor.de/intersystems-documentation/cache-jdbc-driver*

Vendor Name	Details	
	JDBC URL Syntax by Type Server—jdbc:Cache://<server>[:<port>]/<namespace>	**Default Port** 1972
	JDBC Class com.intersys.jdbc.CacheDriver	**JDBC JAR File Name** cachedb.jar

CUBRID

Vendor Name	Details	
Recommended Native Driver		
CUBRID	**Company URL** ***http://www.cubrid.org***	
	Driver URL ***http://www.cubrid.org/?mid=downloads&item=jdbc_driver***	
	JDBC URL Syntax by Type Server—jdbc:cubrid:<server>:<port>:<databaseName>:<username>:<password> :[?<URL attribute>=<value>[&<URL attribute>=<value>] …]	**Default Port** 33000
	JDBC Class cubrid.jdbc.driver.CUBRIDDriver	**JDBC JAR File Name** N/A
	Comments Open source database highly optimized for Web applications.	

Daffodil DB

Vendor Name	Details	
Recommended Native Driver		
Daffodil Software	**Company URL** *http://db.daffodilsw.com*	
	Driver URL *http://sourceforge.net/projects/daffodildb/*	
	JDBC URL Syntax by Type Server—jdbc:daffodilDB://<server>[:<port>]/<databaseName> Embedded—jdbc:daffodilDB_embedded:<databaseName>	**Default Port** 3456 N/A
	JDBC Class in.co.daffodil.db.rmi.RmiDaffodilDBDriver	**JDBC JAR File Name** DaffodilDB_client.jar DaffodilDB_Embedded.jar,

Vendor Name	Details	
	in.co.daffodil.db.jdbc.DaffodilDBDriver	DaffodilDB_Common.jar
	Comments Open source database	

DB2 AS/400

Vendor Name	Details	
Recommended Native Driver		
IBM	**Company URL** *http://www.ibm.com*	
	Driver URL *http://www-03.ibm.com/systems/i/software/toolbox/*	
	JDBC URL Syntax by Type Server—jdbc:as400:// <server>naming=sql;errors=full	**Default Port** N/A
	JDBC Class com.ibm.as400.access.AS400JDBCDriver	**JDBC JAR File Name** jt400.jar
	Shipped with Pentaho Products Pentaho Data Integration	

DB2 Universal Database

Vendor Name	Details	
Recommended Native Driver		
IBM	**Company URL** *http://www.ibm.com*	
	Driver URL *http://www-306.ibm.com/software/data/db2/java*	
	JDBC URL Syntax by Type Server—jdbc:db2://<server>[:<port>]/ <databaseName>[;<URL attribute>=<value>]	**Default Port** 50000
	JDBC Class com.frontbase.jdbc.FBJDriver	**JDBC JAR File Name** frontbasejdbc.jar

Firebird

Vendor Name	Details
Recommended Native Driver	
Firebird Foundation	**Company URL**

Vendor Name	Details	
	http://www.firebirdsql.org	
	Driver URL *http://www.firebirdsql.org/en/jdbc-driver/*	
	JDBC URL Syntax by Type Server—jdbc:firebirdsql:<server>[/<port>]:/<database-file> (JDBC Type 4, official format) Server—jdbc:firebirdsql://<server>[:<port>]/<database-file> (JDBC Type 4, compatibility format) Server—jdbc:firebirdsql:native//<server>[/<port>]:/<database-file> (JDBC Type 2, compatibility format) Server—jdbc:firebirdsql:native://<server>[:<port>]/<database-file> JDBC Type 2, compatibility format. Requires libraries) Embedded—jdbc:firebirdsql:embedded:/<local-database-file> (JDBC Type 2, compatibility format. Requires libraries)	**Default Port** 3050 3050 3050 3050 N/A
	JDBC Class org.firebirdsql.jdbc.FBDriver	**JDBC JAR File Name** jaybird-full-xxx.jar
	Shipped with Pentaho Products Pentaho Data Integration	

FrontBase

Vendor Name	Details	
Recommended Native Driver		
FrontBase	**Company URL** *http://www.frontbase.com*	
	Driver URL *http://www.frontbase.com/cgi-bin/WebObjects/FBWebSite*	
	JDBC URL Syntax by Type Server— jdbc:FrontBase://<host>[:<port>]/<databaseName>	**Default Port** N/A
	JDBC Class	**JDBC JAR File Name**

Vendor Name	Details	
	com.frontbase.jdbc.FBJDriver	frontbasejdbc.jar

Greenplum

Vendor Name	Details	
Recommended Native Driver		
Greenplum	**Company URL** *http://www.greenplum.com*	
	Driver URL *http://jdbc.postgresql.org/download.html*	
	JDBC URL Syntax by Type Server—jdbc:postgresql://<server>[:<port>]/<databaseName>	**Default Port** 5342
	JDBC Class org.postgresql.Driver	**JDBC JAR File Name** postgresql-8.x-xxx.jdbc4.jar
	Comments Greenplum uses the Postgresql JDBC driver	

H2 Database

Vendor Name	Details	
Recommended Native Driver		
H2	**Company URL** *http://www.h2database.com*	
	Driver URL *http://www.h2database.com*	
	JDBC URL Syntax by Type Server—jdbc:h2:tcp://server[:port]/file-path Embedded—jdbc:h2:file-name	**Default Port** 9092 N/A
	JDBC Class jdbc:h2:tcp://server[:port]/file-path org.h2.Driver	**JDBC JAR File Name** h2-x.x.xxx.jar
	Shipped with Pentaho Products • Pentaho Business Analysis Server • Pentaho Data Integration • Pentaho Metadata-Editor Pentaho Report-Designer	
	Comments	

Vendor Name	Details
	Open source Java SQL database

Hive

Vendor Name	Details	
Recommended Native Driver		
Apache	**Company URL** *http://hive.apache.org/*	
	Driver URL N/A	
	JDBC URL Syntax by Type Server—jdbc:hive://<server>[:<port>]/default	**Default Port** 10000
	JDBC Class org.apache.hadoop.hive.jdbc.HiveDriver	**JDBC JAR File Name** hive-jdbc-x.x.x-pentaho-y.y.y.jar Example: hive-jdbc-0.7.0-pentaho-1.0.1.jar*
	Shipped with Pentaho Products • Pentaho Business Analysis Server • Pentaho Data Integration • Pentaho Metadata-Editor Pentaho Report-Designer	
	Comments Data warehouse infrastructure that provides data summarization and ad hoc querying *x.x.x is the Hive version, y.y.y is the Pentaho version. Pentaho has enhanced the standard Hive JDBC driver	

HSQLDB

Vendor Name	Details	
Recommended Native Driver		
HyperSQL	**Company URL** *http://www.hsqldb.org*	
	Driver URL *http://sourceforge.net/projects/hsqldb/*	
	JDBC URL Syntax by Type Server—jdbc:hsqldb:hsql://<server>[:<port>]/<databaseName> Embedded Memory—jdbc:hsqldb:mem:<databaseName>	**Default Port** 9001 N/A N/A

Vendor Name	Details	
	Embedded File — jdbc:hsqldb:file:<database-file>	
	JDBC Class org.hsqldb.jdbcDriver	**JDBC JAR File Name** hsqldb.jar
	Shipped with Pentaho Products • Pentaho Enterprise-Console • Pentaho Business Analysis Server • Pentaho Aggregation-Designer • Pentaho Data Integration • Pentaho Metadata-Editor • Pentaho Report-Designer	

Informix

Vendor Name	Details	
Recommended Native Driver		
IBM	**Company URL** *http://www.ibm.com*	
	Driver URL =	
	JDBC URL Syntax by Type Server—jdbc:informix-sqli://<server>[:<port>]/<databaseName>:informixserver=<dbservername>	**Default Port** 1533
	JDBC Class com.informix.jdbc.IfxDriver	**JDBC JAR File Name** ifxjdbc.jar
	Shipped with Pentaho Products Pentaho Data Integration	

Ingres

Vendor Name	Details	
Recommended Native Driver		
Actian	**Company URL** *http://www.actian.com/*	
	Driver URL *http://esd.actian.com/product/drivers/JDBC/java*	
	JDBC URL Syntax by Type Server—jdbc:ingres://<server>[:<port>]/<databaseName>	**Default Port** 21071
	JDBC Class	**JDBC JAR File Name**

Vendor Name	Details	
	com.ingres.jdbc.IngresDriver	iijdbc.jar
	Comments Open source relational database management system	

InterBase

Vendor Name	Details	
Recommended Native Driver		
Embarcadero	**Company URL** *http://edn.embarcadero.com*	
	Driver URL N/A	
	JDBC URL Syntax by Type Server—jdbc:interbase://<server>/<full_db_path>	**Default Port** N/A
	JDBC Class interbase.interclient.Driver	JDBC JAR File Name interclient.jar
	Shipped with Pentaho Products Pentaho Data Integration	

jTDS Free MS SQL Sybase

Vendor Name	Details	
Recommended Native Driver		
jTDS	**Company URL** *http://jtds.sourceforge.net/*	
	Driver URL N/A	
	JDBC URL Syntax by Type SQL Server—jdbc:jtds:<server_type>//<server>[:<port>][/<database>][;<property>=<value>[;...]]] Sybase—jdbc:jtds:<server_type>://<server>[:<port>][/<database>]	**Default Port** 1433 7100
	JDBC Class interbase.interclient.Driver	**JDBC JAR File Name** jtds-x.x.x.jar
	Shipped with Pentaho Products • Pentaho Enterprise-Console • Pentaho Business Analysis Server	

Vendor Name	Details
	• Pentaho Aggregation-Designer • Pentaho Data Integration • Pentaho Metadata-Editor • Pentaho Report-Designer

LucidDB

Vendor Name	Details	
Recommended Native Driver		
DynamoDB	**Company URL** *http://www.dynamobi.com*	
	Driver URL *http://www.dynamobi.com/c/downloads/stable/*	
	JDBC URL Syntax by Type Server—jdbc:luciddb:http://<server>[:<port>]	**Default Port** 8034
	JDBC Class org.luciddb.jdbc.LucidDbClientDriver	**JDBC JAR File Name** LucidDbClient-x.x.x.jar
	Shipped with Pentaho Products Pentaho Data Integration	
	Comments **Open source BI solution for Big Data**	

MaxDB

Vendor Name	Details	
Recommended Native Driver		
SAP	**Company URL** *http://www.sap.com*	
	Driver URL *http://maxdb.sap.com*	
	JDBC URL Syntax by Type Server—jdbc:sapdb://<server>[:<port>]/<databaseName>	**Default Port** 7210
	JDBC Class com.sap.dbtech.jdbc.DriverSapDB	**JDBC JAR File Name** sapdbc.jar
	Comments Database management system developed and supported by SAP AG	

Mckoi SQL Database

Vendor Name	Details	
Recommended Native Driver		
Mckoi	**Company URL** *http://www.mckoi.com*	
	Driver URL *http://www.mckoi.com/originalmckoisql/index.html*	
	JDBC URL Syntax by Type Server—jdbc:mckoi://<server>[:<port>][/<schema>]/	**Default Port** 9157
	JDBC Class com.mckoi.JDBCDriver	**JDBC JAR File Name** mckoidb.jar
	Comments Open source SQL database written in Java	

Mimer

Vendor Name	Details	
Recommended Native Driver		
Mimer Information Technology	**Company URL** *http://www.mimer.com*	
	Driver URL N/A	
	JDBC URL Syntax by Type Server—jdbc:mimer:<protocol>://<server>[:<port>]/<database>	**Default Port** 1360
	JDBC Class com.mimer.jdbc.Driver	**JDBC JAR File Name** mimer.jar

MonetDB

Vendor Name	Details	
Recommended Native Driver		
MonetDB	**Company URL** *http://www.monetdb.org*	
	Driver URL N/A	
	JDBC URL Syntax by Type	**Default Port** 50000

Vendor Name	Details	
	Server—jdbc:monetdb://<server>[:<port>]/<database>	
	JDBC Class nl.cwi.monetdb.jdbc.MonetDriver	**JDBC JAR File Name** monetdb-jdbc-x.x.jar
	Shipped with Pentaho Products Pentaho Data Integration	
	Comments An open source database system	

MY SQL

Vendor Name	Details	
Recommended Native Driver		
Oracle	**Company URL** *http://www.mysql.com*	
	Driver URL *http://dev.mysql.com/downloads/connector/j/*	
	JDBC URL Syntax by Type Server—jdbc:mysql://<hostname>[,<failoverhost>][:<port>] /<dbname>[?<URL attribute>=<value>[&<URL attribute>=<value>] ...]	**Default Port** 3306
	JDBC Class com.mysql.jdbc.Driver (official class name) org.gjt.mm.mysql.Driver (older class name)	**JDBC JAR File Name** mysql-connector-java-5.x.xx-bin.jar
	Shipped with Pentaho Product • Pentaho Enterprise-Console • Pentaho Business Analysis Server • Pentaho Aggregation-Designer • Pentaho Metadata-Editor • Pentaho Report-Designer	

Neoview

Vendor Name	Details
Recommended Native Driver	
HP	**Company URL** *http://www.hp.com*

Vendor Name	Details	
	Driver URL *https://h20392.www2.hp.com/portal/swdepot/displayProductInfo.do?productNumber=NEO10*	
	JDBC URL Syntax by Type Server—jdbc:hpt4jdbc://<system>[:<port>]/[:][<URL attribute>= <value>[;<URL attribute>=<value> …]	**Default Port** 18650
	JDBC Class com.hp.t4jdbc.HPT4Driver	**JDBC JAR File Name** N/A

Netezza

Vendor Name	Details	
Recommended Native Driver		
IBM	**Company URL** *http://www.netezza.com*	
	Driver URL N/A	
	JDBC URL Syntax by Type jdbc:netezza://<server>[:<port>]/<database>	**Default Port** 5480
	JDBC Class org.netezza.Driver	**JDBC JAR File Name** N/A

OpenBase SQL

Vendor Name	Details	
Recommended Native Driver		
OpenBase International	**Company URL** *http://www.openbase.com*	
	Driver URL *http://www.openbase.com/index.php/products/downloads*	
	JDBC URL Syntax by Type Server—jdbc:openbase://<server>/<databaseName>	**Default Port** N/A
	JDBC Class com.openbase.jdbc.ObDriver	**JDBC JAR File Name** OpenBaseJDBC.jar

Oracle

Vendor Name	Details	
Recommended Native Driver		
Oracle	**Company URL** *http://www.oracle.com*	
	Driver URL *http://www.oracle.com/technetwork/database/features/jdbc/index.html*	
	JDBC URL Syntax by Type Thin Server—jdbc:oracle:thin:@<server>[:<port>]:<sid> OCI Server—jdbc:oracle:oci:@<server>[:<port>]:<sid>	**Default Port** 1521
	JDBC Class oracle.jdbc.driver.OracleDriver oracle.jdbc.OracleDriver	**JDBC JAR File Name** ojdbcx.jar, ora18n.jar
	Comments The OCI server requires OCI libraries	

Pervasive

Vendor Name	Details	
Recommended Native Driver		
Pervasive	**Company URL** *http://www.pervasivedb.com/Pages/default.aspx*	
	Driver URL *http://www.pervasivedb.com/download/Pages/PDBDownloads.aspx*	
	JDBC URL Syntax by Type Server—jdbc:pervasive://<server>[:<port>]/<datasource>	**Default Port** 1583
	JDBC Class com.pervasive.jdbc.v2.Driver	**JDBC JAR File Name** N/A
	Shipped with Pentaho Products • Pentaho Data Integration • Pentaho Report-Designer	
	Comments The data source is the ODBC DSN	

PostgreSQL

Vendor Name	Details	
Recommended Native Driver		
PostgreSQL Global Development Group	**Company URL** *http://www.postgresql.org/*	
	Driver URL *http://jdbc.postgresql.org/*	
	JDBC URL Syntax by Type Server—jdbc:postgresql://<server>[:<port>]/<databaseName>	**Default Port** 5342
	JDBC Class org.postgresql.Driver	**JDBC JAR File Name** postgresql-8.x-xxx.jdbc4.jar
	Shipped with Pentaho Products • Pentaho Data Integration • Pentaho Report-Designer	

SAP DB

Vendor Name	Details	
Recommended Native Driver		
SAP DB	**Company URL** N/A	
	Driver URL *http://www.sapdb.org/sap_db_jdbc.htm*	
	JDBC URL Syntax by Type Server—jdbc:sapdb://<server>/<database_name>	**Default Port** N/A
	JDBC Class com.sap.dbtech.jdbc.DriverSapDB	**JDBC JAR File Name** sapdbc-x.x.x.jar
	Shipped with Pentaho Products Pentaho Data Integration	
	Comments FREE Enterprise Open Source Database	

SQLite

Vendor Name	Details
Recommended Native Driver	
Xerial	**Company URL**

Vendor Name	Details	
	N/A	
	Driver URL *http://www.xerial.org/trac/Xerial/wiki/SQLiteJDBC*	
	JDBC URL Syntax by Type Server—jdbc:sqlite:<filename.db>	**Default Port** N/A
	JDBC Class org.sqlite.JDBC	**JDBC JAR File Name** sqlite-jdbc-x.x.x.jar
	Shipped with Pentaho Products Pentaho Data Integration	

SQL Server

Vendor Name	Details	
Recommended Native Driver		
Microsoft	**Company URL** *http://www.microsoft.com*	
	Driver URL *http://msdn.microsoft.com/en-us/sqlserver/aa937724.aspx*	
	JDBC URL Syntax by Type Server—jdbc:sqlserver:// <server>[:<port>];DatabaseName=<databaseName>	**Default Port** 1433
	JDBC Class com.microsoft.sqlserver.jdbc.SQLServerDriver	**JDBC JAR File Name** sqljdbc4.jar
	Comments The open source jtds driver also works with MSSQL	

Sybase ASE

Vendor Name	Details	
Recommended Native Driver		
SAP	**Company URL** *http://www.sybase.com*	
	Driver URL *http://www.sybase.com/products/allproductsa-z/softwaredeveloperkit/jconnect*	
	JDBC URL Syntax by Type Server— jdbc:sybase:Tds:<server>[:<port>]/ <databaseName>	**Default Port** 5000

Vendor Name	Details	
	JDBC Class com.sybase.jdbc4.jdbc.SybDriver	**JDBC JAR File Name** N/A
	Comments The open source jTDS driver works with Sybase as well	

Sybase SQL Anywhere

Vendor Name	Details	
Recommended Native Driver		
SAP	**Company URL** *http://www.sybase.com*	
	Driver URL *http://www.sybase.com/products/allproductsa-z/softwaredeveloperkit/jconnect*	
	JDBC URL Syntax by Type Server— jdbc:sybase:Tds:<server>[:<port>]/ <databaseName>	**Default Port** 2638
	JDBC Class com.sybase.jdbc4.jdbc.SybDriver	**JDBC JAR File Name** N/A
	Comments This open source jTDS driver works with Sybase as well	

SmallSQL

Vendor Name	Details	
Recommended Native Driver		
SmallSQL	**Company URL** *http://www.smallsql.de/*	
	Driver URL *http://www.smallsql.de/download.html*	
	JDBC URL Syntax by Type Embedded— jdbc:smallsql:databaseName[?URL attribute=value[URLattribute=value] …]	**Default Port** N/A
	JDBC Class smallsql.database.SSDriver	**JDBC JAR File Name** smallsql.jar
	Comments	

Vendor Name	Details
	Java desktop SQL database engine

Teradata

Vendor Name	Details	
Recommended Native Driver		
Teradata	**Company URL** *http://www.teradata.com*	
	Driver URL *http://downloads.teradata.com/download/connectivity/jdbc-driver*	
	JDBC URL Syntax by Type Server—jdbc:teradata://<dbshost>[/<URL attribute>[;<URL attribute>]…]	**Default Port** N/A
	JDBC Class com.teradata.jdbc.TeraDriver	**JDBC JAR File Name** terajdbc4.jar

Vertica

Vendor Name	Details	
Recommended Native Driver		
HP	**Company URL** *http://www.vertica.com*	
	Driver URL TBD	
	JDBC URL Syntax by Type Server—jdbc:vertica://<server>[:<port>]/<databaseName>	**Default Port** 5433
	JDBC Class com.vertica.Driver	**JDBC JAR File Name** N/A

Troubleshooting

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

Version Check

The instructions in this guide are specific to the Pentaho BA Server Enterprise Edition version 4.8-GA. The installation process can change significantly between BA Server releases to address new features, updated requirements, and bug workarounds, so the instructions in this guide should be assumed not to work with any other BA Server version, including the open source BA Server Community Edition.

File Names and Paths

Note: This is the most common installation problem.

Many of the configuration files and paths in this guide are similar, and it is easy to confuse them, which could result in modifying the wrong files or copying to the wrong locations. Double-check your file names and paths and ensure that you've copied all of the right files to all of the correct directories.

Trailing slashes are important; both their inclusion and their absence, depending on the file and parameter or element you are modifying. Follow the examples in this guide exactly unless otherwise directed.

Examining Log Files

If the BA Server fails to start or work properly, the log file you should consult is **pentaho.log** in the `/pentaho/server/biserver-ee/tomcat/bin/` directory. The contents of this file will assist you in tracking down the problem.

Unable to Use the Database Init Scripts for PostgreSQL

The **pg_hba.conf** file contains host-based authentication information. If you can't run the SQL scripts that generate the Hibernate and Quartz databases, it's probably because the default user accounts for each database don't have the right permissions. To change this, edit the file to ensure that connections from local users created by the Pentaho sql scripts (**hibuser** and **pentaho_user**) will be able to connect. The default on Debian-based systems is for local connections yo use **ident** authentication, which means that database users must have local user accounts. In other words, to continue using **ident**, you would have to create local **hibuser** and **pentaho_user** accounts. It's easier to just change the authentication method to something less restrictive, if your IT manager permits you to do so.

context.xml Changes Do Not Take Effect After Re-deploying a WAR

Re-deployment of a WAR or EAR with a custom **context.xml** may cause the original context.xml that you deployed with the original WAR or EAR to become permanently cached. Tomcat in particular will generate a WAR-specific context configuration file, and keep it in place even after the WAR is deleted. The location and naming convention for this file are: **$CATALINA_HOME/conf/Catalina/<host>/<war name>.xml**. Typically this will be something like: **/tomcat/conf/Catalina/localhost/pentaho.xml**. If this file exists, you will have to delete it prior to re-deploying pentaho.war if you have made any changes to context.xml.

JDBC Driver Problems

First, ensure that the correct JDBC driver JARs are installed to the correct locations, then check to make sure that there aren't conflicting driver versions. See the *Pentaho Business Analytics Administrator's Guide* for explanations of how to create JDBC Database connections and driver locations for all parts of Pentaho Business Analytics. Check with your database or driver vendor if you suspect you have having JDBC driver compatibility issues.

Licenses Not Found After Installation

If you've successfully installed the BA Server and Pentaho Enterprise Console, then installed licenses, and are later unable to access Dashboard Designer, Pentaho Analyzer, or Pentaho Interactive Reporting through the Pentaho User Console due to missing licenses, then you must either reinstall those licenses under the user account that will always start the Pentaho Enterprise Console and BA Server services, or you must set a Java virtual machine parameter to point to a static license path. This will override the default license path, which changes depending on which user started the Pentaho Enterprise Console and BA Server services or servers. The parameter is appended to the Tomcat or JBoss service execution commands, and to the Pentaho Enterprise Console **start-pec** script. Examples are shown below:

Windows Tomcat service adjustment command

```
tomcat5 //US//pentahobiserver ++JvmOptions "-Dpentaho.installed.licenses.file=C:
\Pentaho\.installedLicenses.xml"
```

Linux start-pec.sh script snippet

```
"$_PENTAHO_JAVA" -Dpentaho.installed.licenses.file="/home/
pentaho/.pentaho/.installedLicenses.xml" -Xmx512M -XX:PermSize=64M -
XX:MaxPermSize=128M -Djava.awt.headless=true -DCONSOLE_HOME=$DIR_REL
 -Dlog4j.configuration=resource/config/log4j.xml -cp $CLASSPATH
 com.pentaho.pac.server.ProJettyServer
```

Cannot Create Hibernate Tables in MySQL

The Pentaho solution repository uses long text strings that require a longer maximum character limit than the default UTF-8 configuration allows. Therefore, if your MySQL character set is configured to use UTF-8, you must change it to **ASCII** instead in order to use it as a Pentaho solution database. Using UTF-8 will prevent the MySQL initialization scripts from running during installation.

Action Sequences That Call PDI Content Won't Run

If you've established an enterprise repository in PDI to store your jobs and transformations, and you attempt to use that stored PDI content in an action sequence on the BA Server, the action sequence will not execute. This is because the BA Server needs specific connection information for the Data Integration (DI) Server in order to retrieve the job or transformation.

Adding PDI Enterprise Repository Content Support to the BA Server

If you are using a Pentaho Data Integration (PDI) enterprise repository (through a Data Integration Server) to store PDI jobs and transformations, and you plan on using those jobs and transformations in action sequences that will be run on the BA Server, you must install some BA Server plugins from the PDI client tool package. This is not a typical scenario, but there is no harm in performing it if you aren't sure of the details.

1. Download a PDI Enterprise Edition 4.4 client tool archive package from the Pentaho Customer Support Portal.
   The package name (available in both tar.gz and zip formats) is: **pdi-ee-client-4.4.0-GA**
2. Unpack the archive to a temporary location.
3. Edit the `/pentaho/server/biserver-ee/pentaho-solutions/system/kettle/settings.xml` file.
4. Change the value of the **<repository.type>** node from **files** to **rdbms**.
5. Enter your enterprise repository connection information in the proper nodes.
6. Enter the location of your local **repositories.xml** file in the **<repositories.xml.file>** node.

Note: This file is created on your PDI client workstation when you establish a connection to an enterprise repository. Once you have made all of your repository connections on a workstation, copy the **repositories.xml** file to the `~/.kettle/` directory on the BA Server and DI Server machines. If the client tool and servers are all on the same machine, you do not have to copy the file. If you have not yet established any repositories, you will have to revisit this procedure later when your PDI environment is fully configured.

7. Copy the contents of `/data-integration/plugins/` to the `/pentaho/server/biserver-ee/pentaho-solutions/system/kettle/plugins/` directory.

```
cp -r /tmp/data-integration/plugins/* /home/pentaho/pentaho/server/biserver-ee/
pentaho-solutions/system/kettle/plugins/
```

8. Remove the unpacked archive.

```
rm -rf /tmp/data-integration/
```

Your BA Server is now configured to run content stored in the DI Server.

Mac OSX Error When Running Apps

When you run a Pentaho App file on OSX Mountain Lion, you may get a warning message from Gatekeeper. Temporarily changing security settings enables you to open apps successfully. For more information, see *OS X: About Gatekeeper*.

1. **Go to System Preferences** > **Security** > **Privacy** > **General** > **Allow applications downloaded from**.
2. Change the setting to **Anywhere**.
3. Double-click the app.

   A message box appears indicating this app is not signed.
4. Click **Open** to run the app.
5. Reset the security settings to what they originally were.
   The next time you run this app, the message box does not appear and the app runs normally.