pentaho®

open source business intelligence™

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 meet a variety of customer needs. This guide explains how to install the latest release of the Business Analytics Enterprise Edition from pre-configured .zip or .tar.gz packages.

The Pentaho Archive Packages Will Provide	You Must Supply On Your Own
Application server: Tomcat 6.0 **BA Server:** • Pentaho User Console • Analyzer • Dashboard Designer • Interactive Reporting • Pentaho Enterprise Console **Data preparation tools:** • Schema Workbench • Pentaho Data Integration • Metadata Editor **Design tools:** • Report Designer • Design Studio • Aggregation Designer	**A supported operating system:** (See *Compatibility Matrix: List of Supported Products* on page 31 for detailed OS, database, and application server version information.) • Linux • Solaris • Windows • OS X **A Sun JRE version 1.6.0 (6.0)** installed on the server and each workstation that you will install the design or data preparation tools to **A supported database to store solutions in:** • MySQL • PostgreSQL • Oracle **One or more data sources:** • Any JDBC-compliant database • A spreadsheet • A flat file containing comma-separated values

This may not be the best installation path for your production environment. This procedure expects you to provide your own database to store Pentaho Business Analytics solutions, content, and schedules in; and you must use the included Tomcat application server to host the BA Server Web application.

If you would prefer to build the application from source, if you want to use a different solution database or application server, if you plan to implement single sign-on through Central Authentication Service, or if you prefer to use your own Tomcat and MySQL instances, refer instead to the *Pentaho BA Server Manual Deployment Guide*.

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
Step 1	Establish a **PENTAHO_JAVA_HOME** system variable that points to a Sun JRE or JDK version 1.6.0.	
Step 2	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.	
Step 4	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.	
Step 5	Using the included SQL scripts for your database of choice, initialize your Pentaho solution repository.	
Step 6	For Oracle or PostgreSQL solution databases, modify the **context.xml**, **pentaho.xml**, **hibernate-settings.xml**, and **applicationContext-spring-security-hibernate.properties** files appropriately.	
Step 7	If necessary, modify the appropriate XML configuration files to change the default hostname from **localhost**.	
Step 8	If necessary, modify Tomcat's **server.xml** to change the BA Server and Pentaho Enterprise Console port numbers.	
Step 9	If you have Dashboards Enterprise Edition or Pentaho Analysis Enterprise Edition support entitlements, install the Pentaho Dashboard Designer and Pentaho Analyzer plugins by unpacking them to `/pentaho/server/biserver-ee/pentaho-solutions/system/`.	
Step 10	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.	
Step 11	Log into the Pentaho Enterprise Console at `http://localhost:8088` (or whatever address and port you configured), then install licenses; establish users, roles, and data sources; and perform any other necessary configuration information. Use the *Pentaho Business Analytics Administrator's Guide* and the *Pentaho Business Analytics Security Guide* for assistance.	
Step 12	Optionally, modify your Linux or Windows startup configuration so that the BA Server and Pentaho Enterprise Console will start automatically at boot time.	
Step 13	On each client workstation, create an accessible `/pentaho/design-tools/` directory, then download the archive packages for all Pentaho client tools that you have support entitlements for, and unpack them to that directory.	
Step 14	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.	
Step 15	When you are ready to move this server to development or production, remove the Pentaho sample data and solutions.	

Prerequisites

In order to install Pentaho Business Analytics, you should 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. In most installation scenarios, this will require you to be comfortable using your operating system's command line interface and/or graphical system administration tools. You must have the ability to install software, open firewall ports, and start and stop system services on the machine you are installing on. If you don't have the requisite access or permissions, having the cooperation of someone who does (the sysadmin or DBA) is probably good enough.

Installing Pentaho Business Analytics client tools onto end-user workstations is a much simpler process which requires little more than being able to run the graphical installation utility.

Hardware Requirements

Pentaho Business Analytics does not have strict limits on computer or network hardware. As long as you meet the minimum software requirements (note that your operating system will have its own minimum hardware requirements), Pentaho is hardware agnostic but there is a recommended set of minimum system specifications:

Server:

- **RAM:** at least 4GB
- **Hard drive space:** at least 2GB for the software, and more for solution and content files
- **Processor:** dual-core AMD64 or Intel EM64T

Workstation:

- **RAM:** at least 2GB
- **Hard drive space:** at least 1GB for the software, and more for solution and content files
- **Processor:** dual-core AMD64 or Intel EM64T

It's possible to use less capable machines, but in most scenarios this will result in an undesirable level of performance.

Your environment does not have to be 64-bit, even if your processor architecture supports it. All modern desktop, workstation, and server machines have 64-bit processors, but they sometimes ship by default with 32-bit operating systems. Please check with your Information Technology Department if you are unsure of your configuration. If you want to run Pentaho Business Analytics in a pure 64-bit environment, you will have to install a 64-bit operating system, ensure that your solution database and Java Runtime Environment are 64-bit, and install Pentaho Business Analytics via the 64-bit graphical installer, or through the archive-based or manual deployment methods.

Note: A 32-bit JRE has a hard memory limit of 2GB (1.5GB on Windows), so if you have 2GB or more of RAM, you must use a 64-bit JRE on a 64-bit operating system to take full advantage of it. This means that, through architectural limitations, a 32-bit environment will likely be under-powered for a production BA Server deployment.

Important Note on Local User Accounts For Linux/Unix

The section below advises you to create a local user account called **pentaho**. In this new user's home directory you will put the pentaho-solutions directory and store license information recorded by the Pentaho Enterprise Console. You will have to 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.

Caution: 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.

Caution: If you choose to put the pentaho-solutions directory in a parent directory that the pentaho user (or whatever user account you are running your Pentaho services as) does not have write access to, you will not be able to save any reports, analysis views, or dashboards, and you will not be able to set user permissions on any existing content.

Creating a Pentaho System User (Linux/Solaris)

This is for Linux, Solaris, and other Unix-like operating systems only. Windows users do not need to follow this procedure. Theoretically, you can use a different local user account than the one suggested here. However, you will have to modify all of the instructions from here onward 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 in the next section, and 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 fashion those scripts such that they start the service with the **pentaho** user credentials.

How to Check Your Java Version

Pentaho Business Analytics requires a Java Runtime Environment (JRE) or Java Development Kit (JDK) version 1.6 (sometimes referred to as 6.0). Follow the procedure below to see which version of Java is installed on your system and configured to be the default Java executable.

Note: There may be multiple JREs or JDKs on your system, but only one can be set as the global default. Any of these Java instances can be explicitly used for any Java program; if no specific Java executable is called, 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 will also show in the output.

```
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)
```

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 JRE or JDK version 1.6. If you do not set this variable, Business Analytics will use the JRE that the JAVA_HOME variable, which is generally established on most systems that run Java programs, points to. However, this variable may currently be pointing to a newer JRE that is required by a non-Pentaho program and cannot be changed, or it may change automatically in the future due to a Java upgrade and cause unexpected operation in Business Analytics. Therefore, Pentaho recommends that you set PENTAHO_JAVA_HOME as explained below.

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
   ```

The PENTAHO_JAVA_HOME variable is now set, so all Java applications will be able to access the JRE binaries and other resources.

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 and point it to a JRE or JDK version 1.6. If you do not set this variable, Business Analytics will use the JRE that the JAVA_HOME variable, which is generally established on most systems that run Java programs, points to. However, this variable may currently be pointing to a newer JRE that is required by a non-Pentaho program and cannot be changed, or it may change in the future due to a Java upgrade and cause unexpected operation in Business Analytics. Therefore, Pentaho recommends that you set PENTAHO_JAVA_HOME as explained below.

1. In **Windows XP**, right-click on **My Computer**, then select **Properties** in the context menu. In **Windows 7**, 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, and the directory you installed the JRE to into the value field, then click **OK**.
5. In the parent window, click **Apply Changes**.
6. You must restart your computer for the change to take effect.
7. Verify that the variable is properly set.

   ```
   set | %windir%\system32\find "PENTAHO_JAVA_HOME"
   ```

The PENTAHO_JAVA_HOME variable is now set, so all Java applications will be able to access the JRE binaries and other resources.

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. xvfb 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.

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 the Enterprise Edition enablement process.

With the exception of Design Studio, which has separate packages for Linux, OS X, and Windows, all of the below-listed files are platform-agnostic, so there is only one download for all operating systems, databases, and application servers. Most Windows servers do not typically have software to unpack a .tar.gz file, so it is repackaged in .zip format for Windows admins. Other than this archive technology difference, Pentaho Business Analytics packages are identical.

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, and Pentaho Analyzer.

Windows

The packages you will need for a Windows server are:

- **BA Server (includes the Pentaho Enterprise Console):** biserver-ee-4.5.0-GA.zip
- **Pentaho Analyzer plugin:** paz-plugin-ee-4.5.0-GA.zip
- **Pentaho Dashboard Designer plugin:** pdd-plugin-ee-4.5.0-GA.zip
- **Pentaho Interactive Reporting plugin:** pir-plugin-ee-1.1.0-GA.zip

The packages you will need for your Windows workstations are:

- **Report Designer:** prd-ee-3.9.0-GA.zip
- **Metadata Editor:** pme-ee-4.5.0-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:** /4.5.0-GA/client/windows/pad-ee-1.4.0-GA.zip
- **Data Integration:** pdi-ee-client-4.3.0-GA.zip

Linux and Solaris

The packages you will need for a Linux or Solaris server are:

- **BA Server (includes the Pentaho Enterprise Console):** biserver-ee-4.5.0-GA.tar.gz
- **Pentaho Analyzer plugin:** paz-plugin-ee-4.5.0-GA.tar.gz
- **Pentaho Dashboard Designer plugin:** pdd-plugin-ee-4.5.0-GA.tar.gz
- **Pentaho Interactive Reporting plugin:** pir-plugin-ee-1.1.0-GA.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-4.5.0-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-4.3.0-GA.tar.gz

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-4.5.0-GA** archive from wherever you downloaded it to your current location.

   ```
   tar zxvf /home/amenethil/downloads/biserver-ee-4.5.0-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. You must now configure the BA Server for your preferred solution repository database. To do that, you must initialize your database, then potentially add the appropriate database drivers for the included Tomcat application server. Below are instructions for all supported databases. Only follow the sections that apply to your software environment.

Copying Solution Database JDBC Drivers

This is only for Pentaho solution databases. You will also need to install JDBC drivers for your data sources at a later time, but 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 is unable to provide 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 now have the necessary driver to communicate with your solution repository 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 running as a process.

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 will initialize 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 repository 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 via your preferred utility or process.

   There are a variety of ways to execute an SQL script. The simplest way is by using the MySQL client from the terminal.

   ```
   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.

Initializing a PostgreSQL Database

In order to complete this process, you need to have the PostgreSQL native client tools installed, and the database server running as a service. 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 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 not for a specific Pentaho BA Server user; rather it is a generic account that the software uses to update the database when any regular 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

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 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 PostgreSQL configuration:

   Note: 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.

   ```
   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 later in this guide 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 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.

The BA Server is now fully configured to connect to a PostgreSQL solution repository.

Configuring Quartz For PostgreSQL

The default JobStore JDBC database driver does not work properly with PostgreSQL. Follow the below procedure to replace the generic JDBC driver 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 = org.quartz.impl.jdbcjobstore.StdJDBCDelegate`
3. In the aforementioned line, replace the **org.quartz.impl.jdbcjobstore.StdJDBCDelegate** portion 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

Audit logging is enabled by default in the BA Server, but it expects a MySQL database. To configure it for PostgreSQL, follow the below process.

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 and working as intended.

Initializing an Oracle Database

In order to complete this process, Oracle should be installed and running, 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. This connection instance will be called **sampledata**.
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 BI Server to use Oracle as a solution repository with the default Pentaho authentication backend. Even if you will end up using some other 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 later in this guide 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 them to be.

The BA Server is now fully configured to connect to an Oracle solution repository.

Configuring Audit Logging For Oracle

Audit logging is enabled by default in the BA Server, but it expects a MySQL database. 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 now configured for Oracle and working as intended.

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.

Changing the Default Port Numbers

The Pentaho-provided Tomcat instance is preconfigured to run on port **8080**, and the Pentaho Enterprise Console is configured for port **8088**. If you want to change these 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 support entitlement for the products you want to install. Skip this process if you do not have the requisite support entitlements.

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

Note: The Pentaho Analysis Enterprise Edition plugin is separate from Analyzer, and requires a different installation process explained in another part of this guide.

1. If you have not already done so, download the plugin packages from the Pentaho Customer Support Portal.

   The Dashboard Designer package name always begins with **pdd**; the Pentaho Analyzer package always begins with **paz**; the Interactive Reporting plugin always starts with **pir**.
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 register your license files in the Pentaho Enterprise Console, or with the command line license management tool. There are instructions for that process later in this guide. For now, continue on to the next section.

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.

Note: This procedure does not cover Pentaho Analyzer installation.

Note: If you are performing a manual WAR build and deployment, and you want to slipstream 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-3.4.0-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 `/WEB-INF/lib/` directory inside of the deployed pentaho.war:

   - 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 `/WEB-INF/classes/` directory inside of your deployed pentaho.war.
6. 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:

   Refer to *Memcached Configuration Options* for more information on these settings.

   ```
   <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 18.

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.

   There is only one file in this archive: a **wkhtmltoimage** executable.
3. 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/wkhtmltoimage/
   wkhtmltoimage</html-to-image-tool>
   ```
4. 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-->
   ```
5. 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 17.

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 path that you found this utility in, then click **Install**. For instance: **C:\pentaho\biserver-ee\third-party-tools\wkhtmltoimage\**.
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 built and deployed 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 support entitlement 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 `/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 now 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.

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 **mysql** to the name of the init script for your database if it is running on the remote machine, or remove **mysql** 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 mysql
# Required-Stop: mysql
# 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 BI Server Java Xms setting is).
9. In the **Java** tab, set **Maximum memory pool** to **768** (or whatever your preferred BI 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.

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.

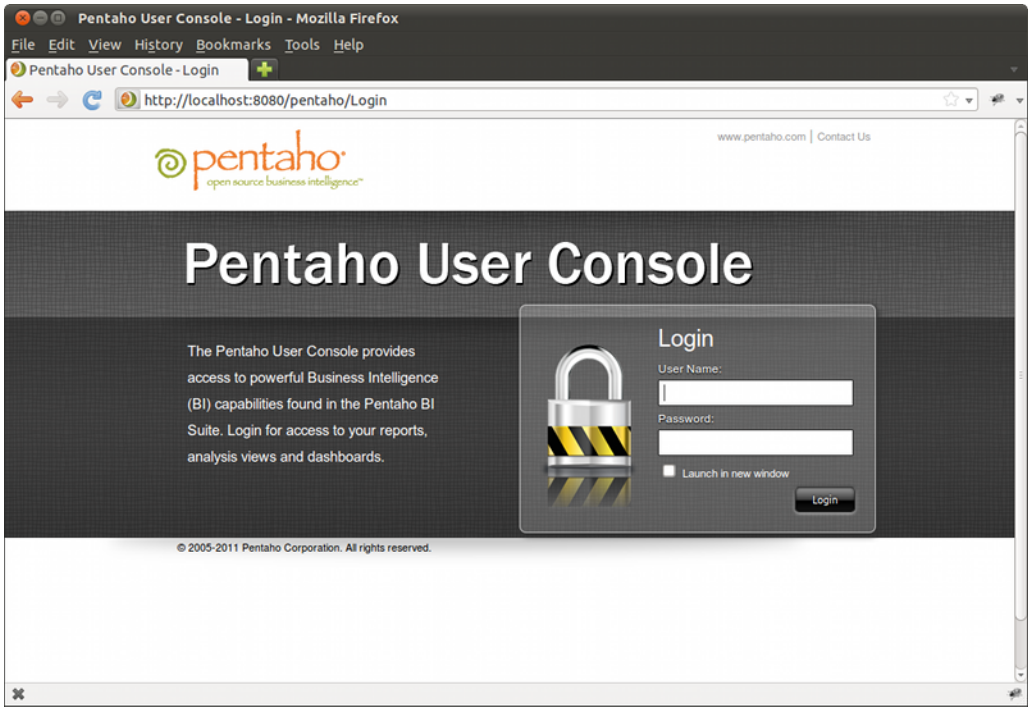
Pentaho Design Studio is typically installed as a complete Eclipse-based application, including the Eclipse development environment. However, if you already have your own Eclipse instance, you can install the Design Studio plugin for Eclipse instead.

1. Exit Eclipse if it is running.
2. If you have not done so already, download the **org.pentaho.designstudio.editors.actionsequence_4.5.0.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.

Testing and Using Your Server

Open a Web browser and navigate to the Pentaho BA Server Web address. If you installed the application onto your local machine, then the URL to reach the reporting and analysis samples is **http://localhost:8080/pentaho**. If you are trying to reach the application from a remote machine, the URL will be **http://example.com:8080/pentaho**, substituting **example.com** with the remote machine's hostname, IP address, or domain name. When the page loads, you should see a Web page similar to the below picture:



Pentaho User Console - Login - Mozilla Firefox
File Edit View History Bookmarks Tools Help
Pentaho User Console - Login
http://localhost:8080/pentaho/Login
pentaho
open source business intelligence
www.pentaho.com | Contact Us
Pentaho User Console
Login
User Name:
Password:
Launch in new window
Login

The Pentaho User Console provides access to powerful Business Intelligence (BI) capabilities found in the Pentaho BI Suite. Login for access to your reports, analysis views and dashboards.

© 2005-2011 Pentaho Corporation. All rights reserved.

You won't be able to log into the Pentaho User Console until you've established user accounts and roles in the Pentaho Enterprise Console.

When your system is properly configured, log in with your administrator account, then verify that you can do the following, assuming your data has been prepared for reporting and analysis:

- Create a new analysis view.
- Create a new ad hoc report.
- If you are a Dashboards Enterprise Edition customer, create a new dashboard with the Dashboard Designer plugin.
- If you are a Pentaho Analysis Enterprise Edition customer, create a new analysis report with the Pentaho Analyzer plugin.
- Schedule a report to run at a regular interval.
- Using Pentaho Report Designer, publish a report to the BA Server.

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 should remove the samples so that they do not get in the way of your BI 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.

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.

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.

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 4.5.0-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 version 4.5.0-stable.

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** will, in some cases, 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. The BA Server installation instructions explain how to add driver JARs to the correct locations; there is also a section titled **Adding a JDBC Driver** in the *Pentaho Business Analytics Administrator's Guide* that explains driver locations for all parts of Pentaho Business Analytics.

Note: Some database vendors (and driver developers) require using JDBC version 4 drivers with a Java 6 environment. 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 ad hoc 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 4.3 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-4.3.0-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.

Compatibility Matrix: List of Supported Products

32-bit Operating Systems

Both servers and workstations are covered in the table below; it's assumed that server software and engines will run on server machines, and design tools will run on workstations. 32-bit operating systems in production environments are becoming more rare. This list will grow shorter with time.

	BA Server/ Enterprise Console	Data Integration Server	Reporting Engine	Analysis Engine	Data Integration Engine	Design Studio	Report Designer	Data Integration	Metadata Editor	Schema Workbench	Aggregation Designer
Windows XP	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Windows 7	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Windows 2008 Server	Y	Y	Y	Y	Y	N	N	N	N	N	N
SUSE Linux Enterprise Desktop/ Server 10	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Red Hat Enterprise Linux 5	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Ubuntu Linux 10.04	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Solaris 10	Y	Y	Y	Y	Y	Y	Y	Y	Y	N	N

64-bit Operating Systems

Both servers and workstations are covered in the table below; it's assumed that server software and engines will run on server machines, and design tools will run on workstations. All modern operating systems have 64-bit versions, and will become the standard in the future. This list will grow longer with time.

	BA Server/ Enterprise Console	Data Integration Server	Reporting Engine	Analysis Engine	Data Integration Engine	Design Studio	Report Designer	Data Integration	Metadata Editor	Schema Workbench	Aggregation Designer
Windows 7	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Windows 2008 Server	Y	Y	Y	Y	Y	N	N	N	N	N	N
Mac OS X	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SUSE Linux Enterprise	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

	BA Server/ Enterprise Console	Data Integratio Server	Reporting Engine	Analysis Engine	Data Integratio Engine	Design Studio	Report Designe	Data Integratio	Metadata Editor	Schema Workbenc	Aggregation Designer
Desktop/ Server 10											
Red Hat Enterprise Linux 5	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Ubuntu Linux 10.04	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Application Server

This refers to the Java application server used by Pentaho server applications and engines.

	BA Server/ Enterprise Console	Data Integration Server	Reporting Engine	Analysis Engine	Data Integration Engine
JBoss 5.1.x	Y	N	Y	Y	Y
Tomcat 6.0.x	Y	Y	Y	Y	Y

Repository

This refers to the solution data stored in the BA Server, DI Server, or in a PDI database repository. This has nothing to do with data source support.

Note: The DI Server has an integrated H2 solution database. Pentaho does not support using other solution databases -- including non-Pentaho H2 database instances -- at this time.

	BA Server/ Enterprise Console	Data Integration Server	Data Integration Engine	Data Integration
DB2	N	N/A	Y	Y
H2 Database	N	Y	Y	Y
Ingres	N	N/A	N	Y
MySQL 5.x	Y	N/A	Y	Y
Oracle	Y	N/A	Y	Y
Postgres 8.x	Y	N/A	Y	Y
SQL Server	N	N/A	Y	Y

Data Sources

All databases that have a **JDBC version 3 driver with full ANSI SQL support** will work with the majority of Pentaho Business Analytics. Pentaho Data Integration includes JDBC drivers for most if its supported data sources. Pentaho Analysis has specific SQL dialect support for certain databases, but will work with others (possibly with poor performance) through its generic JDBC dialect.

	BA Server/ Enterprise Console	Data Integration Server	Reporting Engine	Analysis Engine	Data Integration Engine	Design Studio	Report Designer	Data Integration	Metadata Editor	Schema Workbench	Aggregation Designer
Apache Derby	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
AS400	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Borland InterBase	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Cache	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
DB2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
dBase III, IV, 5	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
EnterpriseDB	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Exasol	N	N	N	N	Y	N	N	Y	N	N	N
Excel	Y	Y	Y	N/A	Y	Y	N	Y	N	N/A	N/A
ExtenDB	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Firebird SQL	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Generic (any JDBC)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Gupta SQLBase	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
H2	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
HSQLDB (formerly Hypersonic)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Informix	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Ingres	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
LucidDB	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
MaxDB	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Microsoft Access	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
MySQL 5.x	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Netezza	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Oracle	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
OLAP (Analysis)	Y	Y	Y	N/A	N	Y	Y	N	N	N/A	N/A
Pentaho Metadata	Y	Y	Y	N/A	N	Y	Y	N	N/A	N/A	N/A
PlumTree	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Postgres	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

	BA Server/ Enterprise Console	Data Integration Server	Reporting Engine	Analysis Engine	Data Integration Engine	Design Studio	Report Designer	Data Integration	Metadata Editor	Schema Workbench	Aggregation Designer
SAP	N	N	N	N	N	N	N	Y	N	N	N
SAP R/3	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
SQL Server 2005	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
sqliteQ	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Sybase	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Teradata	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Text files	Y	Y	Y	N/A	Y	Y	N	Y	N	N/A	N/A
UniVerse Database	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
XML	Y	Y	Y	N/A	Y	Y	N	Y	N	N/A	N/A
XML/A	Y	N	Y	N	N	Y	N	N	N	N	N

Security

This refers to third-party authentication systems that can be integrated with the BA Server, DI Server, and Metadata Editor.

	BA Server/ Enterprise Console	Data Integration Server	Metadata Editor
LDAP	Y	Y	Y
Active Directory	Y	Y	Y
RDBMS	Y	Y	Y
CAS	Y	N	Y
Integrated Windows Authentication	Y	N	Y

Java Virtual Machines

This refers to the Java Runtime Environment (JRE) or Development Kit (JDK) required for running all Pentaho software. **Only the Sun/Oracle distribution is supported**. Ports and derivatives of the standard Oracle distribution (such as FreeBSD Diablo JDK) may work, but are not tested by Pentaho QA and therefore unsupported by Pentaho.

	BA Server/ Enterprise Console	Data Integration Server	Reporting Engine	Analysis Engine	Data Integration Engine	Design Studio	Report Designer	Data Integration	Metadata Editor	Schema Workbench	Aggregation Designer
Sun v 1.6 (6.0)	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Web Browsers

A Web browser is required to use the Pentaho User Console (including Interactive Reporting, Analyzer, Dashboard Designer, Community Dashboard Framework, Data Source Wizard, JPivot, and ad hoc reporting) and Pentaho Enterprise Console. Report Designer requires a Web browser to preview reports exported to HTML format. If you use a browser other than one of those listed here, you will very likely encounter some kind of problem.

	BA Server/ Enterprise Console	Report Designer
Internet Explorer 7, 8, 9	Y	Y
Firefox 3.6, 9, 10	Y	Y
Safari 5.x	Y	Y
Chrome	Y	Y

Legend

Y = Supported

N = Not Supported

N/A = Not Applicable